WOSIS 2004

Eduardo Fernández-Medina,
Julio César Hernández Castro and
Luis Javier García Villalba (Eds.)

Security In Information Systems

Proceedings of the
2nd International Workshop on
Security In Information Systems,
WOSIS 2004
In conjunction with ICEIS 2004
Porto, Portugal, April 2004

INSTICC Press

Eduardo Fernández-Medina,
Julio César Hernández Castro and
Luis Javier García Villalba (Eds.)

Security in Information Systems

Proceedings of the
2nd International Workshop on
Security in Information Systems
WOSIS 2004
In conjunction with ICEIS 2004
Porto, Portugal, April 2004

INSTICC PRESS
Portugal

Volume Editors

Eduardo Fernández-Medina
University of Castilla-La Mancha
Spain
Eduardo.FdezMedina@uclm.es

Julio César Hernández Castro
Carlos III University, Madrid
Spain
jcesar@inf.uc3m.es

and

Luis Javier García Villalba
Complutense University, Madrid
Spain
javiergv@sip.ucm.es

Proceedings of the
2nd International Workshop on
Security in Information Systems – (WOSIS 2004)
Porto, Portugal, April 2004
Eduardo Fernández-Medina,
Julio César Hernández Castro and
Luis Javier García Villalba (Eds.)

Printed in Portugal

ISBN 972-8865-07-4

Foreword

Obtaining a good degree of security in their Information Systems is one of the most pressing challenges facing all kind of organisations today. Although many companies have already discovered how critical information is to the success of their business operations, very few have managed to be effective in keeping their information safe, in avoiding unauthorised access, preventing intrusions, stopping secret information disclosure, etc.

Nowadays, rapid technological advances are stimulating a greater use of information systems in organisations world-wide, which handle large quantities of data, managed by huge databases and datawarehouses. In addition, information systems quite frequently manage information that can be considered sensitive, since it is related to certain intimate or personal aspects of persons (beliefs, medical data, sexual tendencies, etc.) and which must be specially protected.

Many organisations, including not only companies but also governments of several countries, are now realising how security problems can affect both business success and citizen rights, and they are proposing security policies, security planning, personal data protection laws, etc.

All of these, including technological, legislative, ethical and political factors, justifies the importance of secure information systems, and encourage us to research in new techniques, models and methodologies, which could aid designers developing and implanting safe information systems which both protect information and keep within the law. These facts, also, justifies the organization of WOSIS 2004.

The aim of this workshop is to serve as a forum to gather academics, researchers, practitioners and students in the field of Security in Information Systems by presenting new developments, lessons learned from real world cases, and providing the exchange of ideas and discussion on specific areas. From this point of view, the WOSIS 2004 workshop has been a great success, but it would be naïve and pretentious to consider that this success has only been due to their organizers. This is not the case. The organizers of the ICEIS 2004, specially Victor Pedrosa and Joaquim Felipe, have been very helpful and proactive. The invited speakers, Professors Yvo Desmedt and Sushil Jajodia, have contributed a lot to increment the atractiveness and prestige of the WOSIS, helping just by joining us to bring the number of received papers to an overall maximum.

In these conditions, the review proccess has been specially difficult and long, (we have received 53 submissions, of which only 28 papers have been accepted) and it would have been hell if we had not the invaluable help of a very prestigious, competent and flexible Program Committee with the following members: Claudia Barenco from the University of Brazilia, Brazil; Ian Brown from the FIRP & University College London, UK; Sabrina De Capitani di Vimercati and Ernesto Damiani from the Università degli Studi di Milano, Italy; Ed Dawson, from the Information Security Research Center, Queensland, Australia; Markus Dichtl, from Siemens AG, München, Germany; Csilla Farkas from the University of South Carolina, USA; Mariagrazia Fugini from the Politecnico di Milano. Italy; Christian Geuer-Pollmann, of the European Microsoft Innovation Center, Germany; Sushil Jajodia, from George Mason University. USA; Narayana Jayaram from North London University, London, UK; Willem Jonker, of the University of Twente, The Netherlands; Vasilios Katos, of Portsmouth University. UK; Jorge Nakahara form Sao Paulo University. Brazil; Mario Piattini form the University of Castilla-La Mancha, Spain; Jean-Jacques Quisquater of the Université Catholique de Louvain, Belgium; Nicolas Sendrier of INRIA Rocquencourt, France; Simon Shepherd of Bradford University, Bradford, UK; José María Sierra form Carlos III University, Madrid, Spain; Jacques Stern, from the Ecole Normale Supérieure, Paris, France; Robert Tolksdorf of the Freie Universität Berlin, Germany; Ambrosio Toval, of the University of Murcia. Spain; Serge Vaudenay from the Ecole Polytechnique Federale de Lausanne, Swizertland; Duminda Wijesekera from George Mason, USA

We should thank all of them.
We should thank also all the authors who submitted papers to the Workshop, being them accepted or not. The quality was quite high and we must reject some papers of value.

Additionally, the inclusión of a selection of some of the best papers of the Workshop in an special issue of the Information Systems Security Journal, by Auerbach Press, has also contributed to increase the visibility and success of this year's WOSIS. Thanks very much to Rich O'Hanley, the editor, it was a pleasure to work with you.

Finally, we would like to note that we will make our best to repeat this success next year.

The WOSIS 2004 Organizers
Eduardo Fernández Medina Patón
Julio César Hernández Castro
Javier García Villalba

Editors and Co-Chairs

Workshop Chairs

Eduardo Fernández-Medina (Eduardo.FdezMedina@uclm.es)
University of Castilla-La Mancha
Spain

Julio César Hernández Castro (jcesar@inf.uc3m.es)
Carlos III University, Madrid
Spain

and

Luis Javier García Villalba (javiergv@sip.ucm.es)
Complutense University, Madrid
Spain

Program Committee

Claudia Barenco. University of Brazilia, Brazil
Ian Brown. FIRP & University College London, UK
Sabrina De Capitani di Vimercati. Università degli Studi di Milano, Italy
Ernesto Damiani. Università degli Studi di Milano, Italy
Ed Dawson. Information Security Research Center, Queensland, Australia
Markus Dichtl. Siemens AG, München, Germany
Csilla Farkas. University of South Carolina, USA
Mariagrazia Fugini. Politecnico di Milano. Italy
Christian Geuer-Pollmann. European Microsoft Innovation Center, Germany
Sushil Jajodia. George Mason University. USA
Narayana Jayaram. North London University, London, UK
Willem Jonker. University of Twente, The Netherlands
Vasilios Katos. Portsmouth University. UK
Jorge Nakahara. Sao Paulo University. Brazil
Mario Piattini. University of Castilla-La Mancha, Spain
Jean-Jacques Quisquater. Université Catholique de Louvain, Belgium
Nicolas Sendrier. INRIA Rocquencourt. France
Simon Shepherd. Bradford University, Bradford, UK

José María Sierra. Carlos III University, Madrid, Spain
Jacques Stern. Ecole Normale Supérieure, Paris, France
Robert Tolksdorf. Freie Universität Berlin, Germany
Ambrosio Toval. University of Murcia. Spain
Serge Vaudenay. Ecole Polytechnique Federale de Lausanne, Swizertland
Duminda Wijesekera. University George Mason, USA

Table of Contents

Papers

Towards a Systematic Development of Secure Systems

Ruth Breu[1], Klaus Burger[1], Michael Hafner[1], Gerhard Popp[2]

[1] Research Group "Quality Engineering"
Universität Innsbruck, Institut für Informatik
Technikerstraße 13
A - 6020 Innsbruck
Tel.: ++43 - 512 - 507 6114
{Ruth.Breu, Klaus.Burger, m.hafner}@uibk.ac.at
[2] Software & Systems Engineering
Technische Universität München , Institut für Informatik
Bolzmannstraße 3
D-85748 Garching
Tel.: ++49 (89) 289 - 1 78 32
Gerhard.Popp@in.tum.de

Abstract. In this paper we outline a new process model for security engineering. This process model extends object oriented, use case driven software development by the systematic treatment of security related issues. We introduce the notion of security aspects describing security relevant requirements and measures at a certain level of abstraction. We define a micro-process for security analysis supporting the systematic development of secure components within iterative systems development.

1 Introduction

Due to the increasing number of distributed applications security plays a more and more important role within systems development. In particular, evolving new web technologies supporting the dynamic interconnection between software components and novel mobile devices require a high level of security.

Today's process models like the Unified Process ([1, 2]) or Catalysis ([3]) treat security aspects as non-functional requirements among others. Our claim is that security is a requirement which has to be considered in all stages of development and which needs particular modelling techniques to be captured.

Moreover, the development of secure systems poses particular challenges to the development process. This comprises the separation of requirements and measures, the traceability of security requirements, the correctness of the measures taken and the completeness of requirements and measures.

Observing that security relevant issues are often merely considered at the technical level (by using encryption techniques, security protocols, logging etc.) our main goal is to separate abstraction levels and to specify requirements and measures at the appropriate level. This ranges from eliciting security requirements in the business model, taking into account security specific aspects at the level of work processes to the technical level of the software architecture. A first step towards this aim has been achieved in [4, 5, 6, 7].

We describe both security requirements and measures in the artefact of the core process and call these requirements and measures *security aspects*. Our core process is based on a general approach which can be easily mapped to any of the established process models [8]. The core artefacts are the Business Model (describing work processes), the System Requirements (describing the system's use cases), the Application Architecture (describing the system's logical components and the core message flows) and the Software Architecture (describing the technical structure).

A special challenge to the development process of security critical systems is imposed by the concept of iterative software construction. For instance, the introduction of classes in a new increment requires an elaboration of the access rights which in turn may lead to new measures and use cases (e.g. logging the access to the new objects and surveying this logging information).

We meet this challenge by introducing a micro-process for security analysis. The micro-process comprises the five steps of security requirements elicitation, threats and risk analysis, taking measures and the correctness check relating measures and requirements. These five steps are repeatedly performed at each level of abstraction during the incremental development.

The two steps of threats and risk analysis support the transition from requirements to measures by gathering the potential threats related to the security requirements and by estimating the occurrence of each threat and its potential harm. Separating the application and the technical level we define two new artefacts - the Application Risks and the Technical Risks containing the description of threats and risks at the respective level of abstraction.

The structuring of this paper is as follows. In section 2 we outline the principles of the design process our approach is based on. Section 3 presents the activities and artefacts of our process model and in section 4 a conclusion is drawn.

We illustrate this process model with a case study based on "TimeTool", a software project which was realized at the University of Innsbruck. TimeTool is a software package supporting project controlling and administration. Based on a three tier architecture it was implemented on top of the J2EE platform. The application is accessed through a web front-end. Team workers' working time is calculated in real time through a log in / log out timestamp, with the option of performing the entries manually under special circumstances. The system performs specific checks (on date and booked time) automatically and offers administrative and controlling features to the project manager (team worker management, statistical reports generation).

2 Basic Concepts of an Object Oriented Software Process

In this section we present the key concepts of the core object oriented process.

The **Business Model** captures the organizational environment of the IT-system. It describes the *actors*, the *activitie*s and the *objects*. In TimeTool the actors are the *project manager*, the *team worker* and the *administrator*. Example activities are *Book Worked Hours* and *Post Adjustment.* Example objects in the application domain are the *project*, the *booking* and the *team worker.*

Actors, activities and objects are modelled in activity diagrams and class diagrams. In its system view the business model focuses on the work processes and is independent of the IT-system.

The **System Requirements** consist of the use case diagram and the class model and give a black box view of the system. A sample use case is *Book Worked Hours*.

Commonly, the class model of the System Requirements is a refined version of the class model of the Business Model. The textual description of a use case comprises sections for pre- and post-conditions of the use case, for the main steps and interactions when performing the use case and sections for exceptions and variants.

The **Application Architecture** refines the level of description. The system is divided into a set of *logical components*. Each component is responsible for a portion of the system structure and behaviour. It consists of component diagrams, a set of sequence diagrams and state diagrams. Interfaces enable the independent development of the system components. Textual descriptions of the use cases are refined into *scenarios*, describing the use cases as message flows between objects.

Besides the artefacts themselves, their sequence and interdependencies form the main characteristics of a process. In this respect, iterative development [2] is one of the most important concepts in modern process models. For instance, for the System Requirements this means, that not all use cases are specified in detail in a first step but only the kernel ones, and other use cases are specified in later stages of design.

3 Core Concepts of a Process Model for Security Engineering

In this section we present the concepts of our process model. The core idea is the introduction of a micro-process which we call *Security Analysis*. In section 3.1 we clarify the basic idea of the micro-process and its integration in the core process. Section 3.2 is devoted to the security enhanced artefact.

3.1 The Security Analysis Process

Security related aspects in the software lifecycle are tackled in a five step approach which we call *Security Analysis* (Table 1).

We illustrate these five steps by the scenario in Table 2. In our process model we treat the Security Analysis as a micro-process which is *performed* at *each level of abstraction* and *for each increment*. This has the following advantages:

- Requirements and measures are each explored and described at the appropriate level of detail. Each security requirement can be traced along the levels of abstraction. More precisely, each requirement is transformed into one or several requirements or into some measure at the abstraction level beneath.

Table 1. The Security Analysis Process

1. *Security Requirements Elicitation* – Specify security requirements in the context of the core artefact.
2. *Threats Modelling* – Gather potential threats related with the security requirements.
3. *Risk Analysis* – Estimate the occurrence of every threat and its potential harm either quantitatively or qualitatively. This provides the basis for the decision whether a threat has to be countered or not.
4. *Measures Design* – Design appropriate measures taking into account the result of the risk analysis and integrate the description of these measures into the core artefacts.
5. *Correctness Check* – Check the chosen measures (formally or informally) against the specified requirements and decide what requirements still wait for realisation.

Table 2. Scenario of a Security Analysis

1. The non-repudiation of the activity *Adjustment Posting* (performed either by the project manager or the team worker) is identified as a security requirement in the business model of "TimeTool".
2. This activity is associated with the following threats:
   -The team worker performs a positive time adjustment on her account in order to increase her billable time.
   - The project manager performs negative time adjustments on several accounts in order to hide budget overruns.
3. The probability of occurrence is estimated as high, the possible damage is estimated as substantial.
4. The measures to counter the threats involve both the business level and the IT System. On the one hand side the business process is reorganised and improved, e.g. adjustment postings require additional information and involved parties are automatically notified. On the other hand a functional requirement, namely that all *Adjustment Postings* have to be logged by the system is added to the System Requirements.
5. The proposed measures are checked against the requirement of non-repudiation. The result of this new requirement is that the involved parties must not have access to logging information.

Security Aspects. Security measures at one level of abstraction may be seen as security requirements at a lower level of abstraction. This is why we generalise requirements and measures to the concept of *security aspects*.

Security aspects are security relevant parts described in the core artefact. For expressing some of the security aspects we introduce extended notation techniques, e.g. in the Business Model. Other security aspects can be described within the UML notation (e.g. sequence or state diagrams can be used to describe security protocols).

A core idea in our approach is that we relate security aspects by a *realisation relation*. Each security aspect is realised by zero, one or many other security aspects. For instance, the security aspect of non-repudiation in the Business Model of the sample scenario of Table 2 is realised by a business process enhancement and a logging mechanism. Aspects realised by no other aspect are measures at the lowest level of abstraction. In general, the realisation relation may relate aspects at different levels of abstraction (in different core artefacts) or at the same level of abstraction. The relation is many-to-many. Our approach has the advantage that security requirements and measures can be traced through several artefact. This supports a systematic check for correctness and completeness.

Artefacts and their Integration into the Process. The enhanced process involves the following artefacts:

- All core artefacts are also part of the enhanced process. We extend these core artefacts by techniques and methods to express security requirements at the given level of abstraction.
- We define two additional artefacts – the *Application Level Risks* and the *Technical Risks* – documenting threats and risks at the application and platform-dependent level.

Table 3. Iterations of a Given Core Artefact

Step 0:	Develop the core artefact as described in the core process.
Step 1:	Enhance this artefact by security aspects modelling requirements according to given methodological guidelines.
Step 2, 3:	Analyse related threats and risks in the respective Risks artefact.
Step 4:	Integrate security aspects in the core artefact modelling measures. These security measures may refer to security aspects of the abstraction level above or of the same abstraction level. Document the realisation relation between the new aspect and the given aspects.
Step 5:	Perform the correctness check and eventually add refined security aspects to be fulfilled by the abstraction level beneath.

The Risks artefacts complement the specification of requirements by supporting the choice of appropriate measures and the definition of test cases. In the Risks artefacts both external and internal risks are analysed. From the viewpoint of a given core artefact the Security Analysis results in the series of steps (for each increment) depicted in Table 3.

A crucial aspect in this method is the comprehensive specification of security requirements. In our approach we provide systematic checks of the base models with respect to the following settled set of objectives:

- Confidentiality – keeping content from all but those authorised to have it
- Authenticity – establishing the validity of transmission, message or originator
- Data Integrity – prevention of unauthorised modification of data

- Non-Repudiation - guarantee that an entity cannot deny previous commitments of actions
- Availability – ensuring that unauthorised subjects cannot prevent authorised ones from the execution of their functions

Figure 1 summarises the activities and artefact of the enhanced process. In the subsequent section we will demonstrate the application of our process in more detail focusing on the security enhancements of the core artefact.

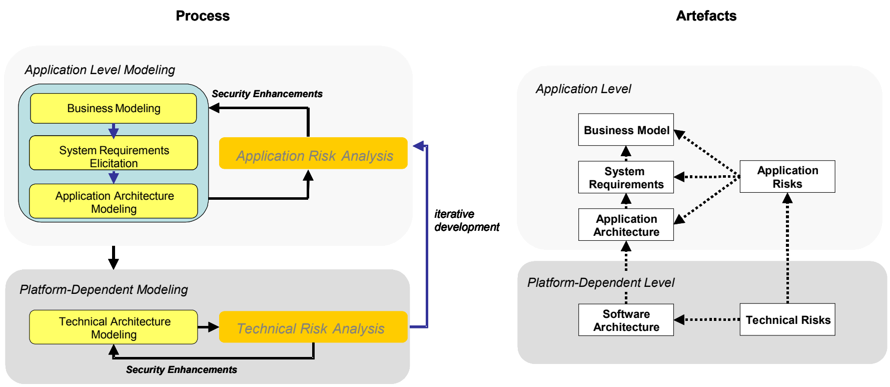


Figure 1. The Security Process Model

3.2 Security Enhanced Core Artefacts

In the following we briefly describe for each artefact how security requirements are specified, what kind of threats are captured in the correlated Risks document and what kind of measures the core artefact may comprise. In our approach we follow a schematic pattern-based elicitation of security requirements. This may be complemented by textual or formal specifications. For a more detailed presentation we refer to related publications ([9, 10]).

Business Model. The systematic security check of the given business model comprises the following aspects.

Confidentiality is specified at the business level at a rough level of detail in the class diagram. Each class (or attribute) is provided with one of the keywords public, confidential or secret. Another aspect is related with the object flow between activities. Object flows between different actors require some data exchange. For every single object flow in the business process model we have to check if confidentiality of the object involved is critical. Data Integrity concerns the access to objects and is captured by the security levels in the class diagram described above. Moreover, analogously to above, each object flow in the process model is checked for the aspect of data integrity.

As an example, think of the workflow of an adjustment posting (Figure 2). Since the adjustment posting and its confirmation as well as the generated notification contain the data about billable time both, their confidentiality and integrity is critical.

Authenticity is a requirement which refers to activities and actors in the business model. Each activity has to be checked if authenticity of the executing actor is a critical requirement. In our example the project manager has to be authenticated when executing the activities *confirm adjustment* and, like any team worker, *perform booking* (of his billable time spent on project).

Non-repudiation is again a requirement involving activities and actors. For each activity it has to be checked if it is important that the executing actor cannot repudiate the execution of this activity. In our case study the activity *Post Adjustment* is associated with the requirement of non-repudiation.

Each of the security requirements related threats, like the one described in Figure 2 (item 2), has to be described in the Application Level Risks document and estimated. Measures at the level of the business model may involve the reorganisation of work processes typically including the separation of duties and the way of handling objects (e.g. requiring the destruction of certain documents after use).

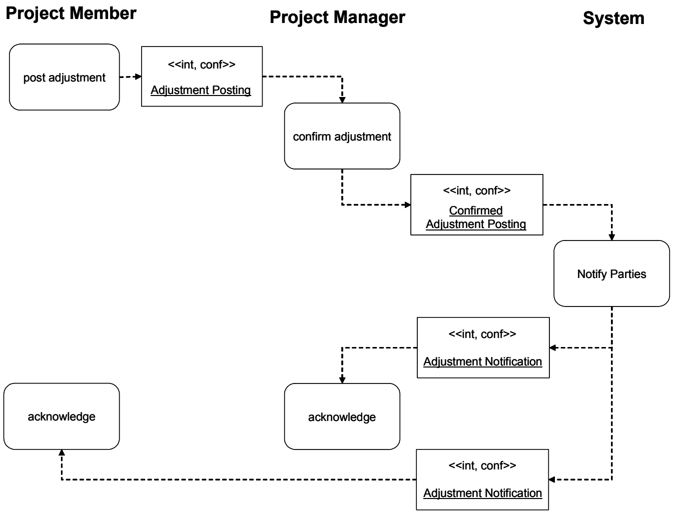


Figure 2. Sample Object Flow with Security Requirements

System Requirements. In the System Requirements document the security requirements of the business model are systematically detailed and put into the context of the use cases and the extended class diagram.

An important activity within the specification of system requirements is the development of a detailed *access policy*. In [11] we present a formally based model for specifying access rights in the context of class diagrams and use case diagrams. Using an extension of OCL predicates (or informal text) the model describes for each actor and each class (or, on a more detailed level, for each method or attribute) the kind of permission. As examples we specify:

The team worker has read access to his own accounted working hours.
The team worker has write access on his own accounted working hours whose status I not set to "frozen".
The project manager has read access to the accounted hours of all projects members of his project.

Here reading and writing access means an indirect access through application of the use cases. The access policy is developed in cooperation with customers and/or end-users. If the access rights are specified formally they can be automatically transformed into code. For a more detailed presentation of our specification framework we refer to [11].

Other security relevant aspects are part of the textual description of use cases or are treated in new use cases (such as the use case *log in* in which the authentication takes place). The schematic textual description of use cases is now extended by a section **security,** describing the enhanced security aspects.

As an example, the textual description of the use case *Adjustment Posting* is enhanced by the security aspects A1 and A2 as shown in Table 4. Both requirements are refined versions of the security requirements in the business model.

Table 4. Security Section of the Use Case Adjustment Posting

use case Adjustment Posting
… (previous textual description of the use case)
security
A1 The adjustment posting is logged by the system.
A2 The team worker has to be authenticated before starting the use case.
A3 Web browser and TimeTool have to authenticate each other before the transaction starts.
A4 The system must guarantee the confidentiality and integrity of the input data.
A5 The use case must be available during extended working hours (6a.m. to 22a.m.) with a maximum of 2 continuous working days breakdown per month.

Further security requirements contained in the use case description analyse the communication with external systems. In our example the use case *Adjustment Posting* involves the communication with the web browser of the client. Since confidential data is sent across the network, requirements A3 and A4 were added to the security section on the present use case.

Finally, another requirement which comes into play at this level of abstraction is availability (A5) which guarantees a minimum availability of the system during working hours and days.

The security analysis of the use cases in general leads to new threats (like the threat that a team member tries to manipulate the logging information). These threats have to be integrated in the Application Level Risks document and should be linked with the relevant parts in the System Requirements.

Application Architecture. Based on the security requirements stated in the System Requirements, the security enhancement of the Application Architecture mainly deals with the design of appropriate measures. This comprises the following steps:

- Definition of logical security components and their interfaces
- Extension of the sequence diagrams describing the message flow of use case execution by security specific messages

Concerning the security requirements of the use case model the development of the (Security) Application Architecture involves the following kind of measures:

- Design of authentication procedures
- Access control and access rights management at the application level
- Error tracing measures (e.g. for authentication and access control)
- Introduction of security protocols for data integrity, confidentiality and non-repudiation

As an example, the use of a challenge-response protocol for authentication or of blind signatures for data integrity is chosen at this level of abstraction. The check of correctness of these measures against the requirements may require complex mathematical proofs.

For this reason and for the systematic transition from requirements to measures the use of *security patterns* is of great importance in this stage of design. Catalogs and classifications of security patterns currently are developed by a number of groups [12, 13, 14, 15]. However, the flexible integration of security patterns into concrete models still poses a number of unsolved problems, e.g. concerning the dynamic behaviour of the resulting system and the combination and interference of patterns.

Threats recovered at the level of the Application Architecture relate to the chosen measures. An example for this are the threats created when applying the still very popular perimeter security model of the “mainframe era” to a distributed server environment [16].

Software Architecture. This phase starts with the design of the basic architecture (comprising the network structure, database structure, choice of programming languages, frameworks and so on) and the logical components which are distributed across network nodes.

The security enhancement to the Software Architecture (Security Software Architecture for short) is then developed in five consecutive steps as listed in Table 2.

After the mapping of the security requirements stated in the System Requirements and the Application Architecture, the technical threats of this basic architecture are analysed. The technical threats can be gathered independently of the application domain e.g. based on checklists [17, 18, 19]. Examples for technical threats in a concrete architecture are wiretaps, insider abuse of net access or system penetration. After being identified each technical threat is related with the application-level threats cross-checking the technical and the application-dependent level. The third step leads to an estimation of associated risks (taking into account the technical opportunities of the attacker and the possible damage).

In the fourth step the security architecture is designed. Typically this comprises the following aspects:

- realisation of the security measures described in the Application Architecture on the chosen platform
- selection of special hardware devices (such as smart cards, key generators)
- access control of databases
- choice of predefined components or frameworks supporting security (like PGP, J2EE, etc.)
- introduction of measures ensuring the availability of system services and disaster recovery

Finally, the proposed measures have to be evaluated in their compliance with the requirements defined at the beginning of the phase.

For the development of the Software Architecture the security of the basic environment has to be taken into account as well. If the system runs on a platform which provides basic protection (e.g. through firewalls, intrusion detections, virus scanners) the related aspects do not have to be analysed in the context of the project. In the other case the basic protection has to be provided as part of the systems development.

Concerning the systematic transition from the Application Architecture to the Software Architecture the use of frameworks or model driven approaches is the primary choice. Their goal is the realisation of systems in a platform-independent style following the idea that the framework takes over the platform-dependent part of the work. A prominent exponent of such approaches is the J2EE-environment [20]. In SECTINO we develop a framework for developing secure workflows based on Web Services [25].

4 Conclusion

In the preceding sections we sketched a process model supporting the systematic development of security-critical systems within the framework of object oriented use case based modelling. Security analysis is integrated via a micro-process specifying a series of consecutive steps, which are repeatedly applied and refined through all the stages of the design process.

The comprehensive view of the whole design process across all layers of abstraction and its rigorous support of traceability distinguishes our approach to the development of secure systems from related work, e.g. [6, 21, 22, 23, 24, 26, 27, 28]. Our process model aims to integrate these existing approaches to security engineering.

Our goal is to develop a tool-supported process appropriate for industrial use. Positive results from pilot projects in an industrial context encourage us to move further to this direction. Currently, the process model is extended and elaborated by our groups in many directions, ranging from the formal modelling of access policies, the tool-supported management of requirements, threats and risks to the platform-independent development of security solutions to component framework (J2EE / .NET) related security issues.

References

[1] I. Jacobson, G. Booch, J. Rumbaugh: *The Unified Software Development Process*. Addison-Wesley, 1999.

[2] P. Kruchten: *The Rational Unified Process*. Addison-Wesley, 1999.

[3] D. D´Souza, A. Wills: *Components and Frameworks with UML – The Catalysis Approach*. Addison-Wesley, 1999.

[4] R. Breu, K. Burger, M. Hafner, G. Popp, J. Jürjens, G. Wimmel: *Security-Critical System Development with Extended Use Cases.* Accepted for APSEC03.

[5] D. Basin, J. Doser, T. Lodderstedt: *Model Driven Security for Process-Oriented Systems*. In *8th ACM Symposium on Access Control Models and Technologies*. ACM Press, 2003.

[6] D. Firesmith: *Security Use Cases*. In: Journal of Object Technology 2(3), 2003. http://www.jot.fm/issues/issue_2003_05/column6

[7] T. Lodderstedt, D. Basin, J. Doser: *Secureuml: A uml-based modeling language for model-driven security.* In: J.-M. Jézéquel, H. Hussmann, S. Cook (eds.): UML 2002. Lecture Notes in Computer Science, vol. 2460, Springer, 2002.

[8] www.v-modell.iabg.de

[9] R. Breu, K. Burger, M.Hafner, G. Popp: *Core Concepts of a Process Model for Security Engineering*. Accepted for Icssea 2003.

[10] G. Popp: *Vorgehensmodelle für die Entwicklung sicherer Systeme*. Dissertation, Munich University of Technology, to appear.

[11] R. Breu, G. Popp: *Actor-Centric Modeling of Access Rights.* Submitted for publication.

[12] J. Yoder, J. Barcalow: *Architectural Patterns for Enabling Application Security* . 4th Conference of Pattern Languages of Programs (PloP), 1997.

[13] E. Fernandez, R. Pan: *A Pattern Language for Security Models.* 8th Conference of Pattern Languages of Programs (PloP), 2001.

[14] B. Blakley: *Securtiy Design Patterns.* The OpenGroup. 2002. http://www.opengroup.org/security/gsp.htm

[15] M. Schumacher: *Security Engineering with Patterns.* PhD Thesis, Lecture Notes in Computer Science, LNCS 2754, Springer, 2003.

[16] M. Kis: *Information Security Antipatterns in Software requirements Engineering*. 9th Conference of Pattern Languages of Programs (PloP), 2002.

[17] J.D. Meier et al.:*, Improving Web Application Security, Threats and Countermeasures*. Microsoft Corporation, 2003.

[18] Bundesamt für Sicherheit in der Informationstechnologie: *IT Baseline Protection Manual.* Bonn, 2001. http://www.bsi.de/gshb/english/menue.htm

[19] T. R. Peltier: *Information Security Risk Analysis.* Auerbach, 2001.

[20] http://java.sun.com/j2ee/

[21] R. Anderson: *Security Engineering*. John Wiley, 2001.

[22] J. Jürjens: *Secure Systems Development with UML*. Springer, to appear.

[23] G. Sindre, A. Opdahl: *Templates for misuse case description*.In: Proc. Seventh International Workshop on Requirements Engineering: Foundation of Software Quality (REFSQ'2001), 2001.
[24] E.B. Fernandez, J.C. Hawkins: *Determining Role Rights from Use Cases*. Proc. ACM Workshop on Role-Based Access Control. Proceedings of the second ACM workshop on Role-based access control, United States, 1997.
[25] R. Breu, M. Hafner, B. Weber: *Modeling and Realizing Security-Critical Inter-Organizational Workflow.* Submitted for Publication.
[26] G. Sindre, G. G. Firesmith, A. L. Opdahl: A Reuse-Based Approach to Determining Security Requirements. In: Proc. 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03), Klagenfurt/Velden, Austria, 2003.
[27]A. Toval, A. Olmos, M. Piattini: *Legal Requirements Reuse: A Critical Success Factor for Requirements Quality and Personal Data Protection*. In: Proc. IEEE Joint International Conference on Requirements Engineering (RE'02) Essen, Germany, 2002.
[28]J. A. Toval, J. Nicolás, B. Moros, F. Garcia: *Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach.* Requirements Engineering (6), London, 2002.

Group Hierarchies with Constrained User Assignment in Linux

Gail-Joon Ahn[1] and Seng-Phil Hong[2]

[1] University of North Carolina-Charlotte, Charlotte, NC, USA
gahn@uncc.edu
[2] LG-CNS, Seoul, Korea
philhong@lgcns.com

Abstract. In this paper we investigate one aspect of RBAC administration concerning assignment of users to roles. A user-role assignment model can also be used for managing user-group assignment. We overview a constrained user-group assignment model and describe its implementation in the Linux system. Rather than set user and file rights individually for each and every user, the administrator can give rights to various groups, then place users within those groups in Linux. Each user within a group inherits the rights associated with that group. We describe an experiment to extend the Linux group mechanism to include group hierarchies and decentralized user-group assignment can be implemented by means of setgid programs.

1 INTRODUCTION

Role-based access control (RBAC) has received considerable attention as a promising alternative to traditional discretionary and mandatory access controls (see, for example, [NO95,FCK95,GI96,SCFY96,JGAS01]). In RBAC permissions are associated with roles, and users are made members of appropriate roles thereby acquiring the roles' permissions. This greatly simplifies management of permissions. Roles are created for the various job functions in an organization and users are assigned to roles based on their responsibilities and qualifications. Users can be easily reassigned from one role to another. Roles can be granted new permissions as new applications and systems are incorporated, and permissions can be revoked from roles as needed.

Sandhu and Bhamidipati [SB97] introduced the URA97 model for decentralized administration of user-role membership (URA97 stands for user-role assignment 1997). They simply focused on user-role assignment without consideration of the important constraints such as separation of duty (SOD) constraints. An example of SOD policy may be “the patent submitted to a patent authorization agency can be reviewed only by a member of its patent review committee.” This simple role-based access control may not be adequate for expressing many business policies. An example of such policy is “none of the applicants of the patent is eligible to review a patent, even though the applicant is a patent review committee member.” These policies, also known as SOD constraints should be dealt with user-role assignment.

Constraints are an important aspect of RBAC and are often regarded as one of the principal motivations behind RBAC. Although the importance of constraints in RBAC has been recognized for a long time, they have not received much attention. [AS00] recently showed that role-based authorization constraints can be expressed by the specification language called RCL 2000. We use the concept of static separation of duty (SSOD) borrowed from this work. The central contribution of this article is to describe how we can achieve this kind of constraints during user-role assignment named constrained user-role assignment as an extension of URA97.

A user-role assignment model can also be used for managing user-group assignment and therefore has applicability beyond RBAC. The notion of a role is similar to that of a group, particularly when we focus on the issue of user-role or user-group membership. For our purpose in this paper we can treat the concepts of roles and groups as essentially identical. The difference between roles and groups was hotly debated at the ACM Workshop [YCS95,San97]. There exists the consensus that a group is a named collection of users (and possibly other groups). Groups serve as a convenient shorthand notation for collections of users and that is the main motivation for introducing them. Roles are similar to groups in that they can serve as a shorthand for collections of users, but they go beyond groups in also serving as a shorthand for a collection of permissions. Assigning users to roles or users to groups are therefore essentially the same function.

The rest of the paper is organized as follows. In section 2, we review the URA97 grant model. Section 3 discusses role-based authorization constraints. Section 4 describes constrained user-group assignment (CONUGA) including implementation details. Section 5 concludes the paper.

2 OVERVIEW OF URA97 MODEL

This section reviews URA97. We often use the term group as an identical notion of role. Our description of URA97 is informal and intuitive. A formal statement of URA97 is given in [SB97]. In this section we s imply give a quick overview of the grant model which is dealing with granting a user membership in a group.

2.1 User-Group Grant Model

URA97 imposes restrictions on which users can be added to a group by whom. URA97 requires a hierarchy of groups (such as in Figure 1) and a hierarchy of administrative groups (such as in Figure 2). The set of groups and administrative groups are required to be disjoint. Senior groups are shown toward the top and junior ones toward the bottom. Senior groups inherit permissions from junior groups. We write $x > y$ to denote x is senior to y with obvious extension to $x \geq y$. The notion of prerequisite condition is a key part of URA97. User-group assignment is authorized in URA97 by the *can-assign* relation.

Definition 1. *A **prerequisite condition** is a boolean expression using the usual $\wedge$ and $\vee$ operators on terms of the form x and $\overline{x}$ where x is a regular role (i.e., $x \in R$). A prerequisite condition is evaluated for a user u by interpreting x to be true if $(\exists x' \geq x)(u, x') \in UA$ and $\overline{x}$ to be true if $(\forall x' \geq x)(u, x') \notin UA$. For a given set of roles R let CPR denotes all possible prerequisite conditions that can be formed using the roles in R.*

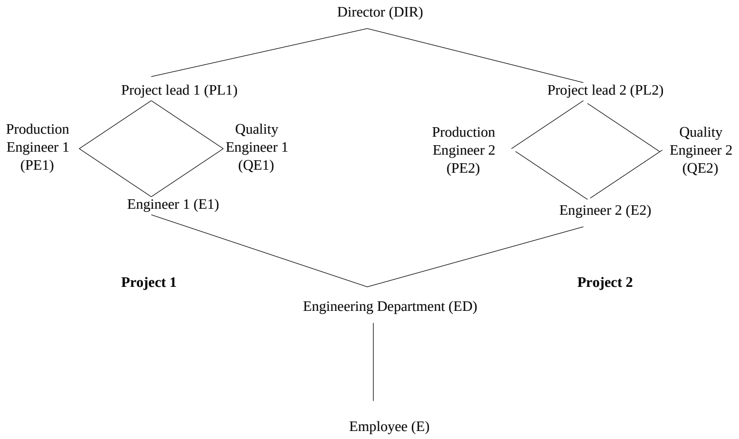


Fig. 1. An example group hierarchy

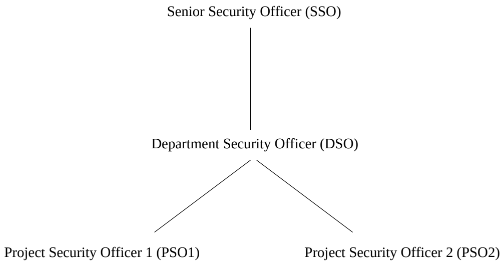

Senior Security Officer (SSO)
Department Security Officer (DSO)
Project Security Officer 1 (PSO1)
Project Security Officer 2 (PSO2)

Fig. 2. An example administrative group hierarchy

Definition 2. *The URA97 model controls user-role assignment by means of the relation* can-assign $\subseteq AR \times CPR \times 2^R$.

The meaning of *can-assign*$(x, y, \{a, b, c\})$ is that a member of the administrative role x (or a member of an administrative role that is senior to x) can assign a user whose current membership, or non-membership, in regular roles satisfies the prerequisite condition y to be a member of regular roles a, b or c.

2.1.1 Range Notation

URA97 also defines *can-assign* by identifying a range within the role hierarchy by means of the familiar closed and open interval notation.

Definition 3. *Role sets are specified in the URA97 model by the notation below*

$$\begin{aligned}
[x, y] &= \{r \in R \mid x \geq r \wedge r \geq y\} \\
(x, y] &= \{r \in R \mid x > r \wedge r \geq y\} \\
[x, y) &= \{r \in R \mid x \geq r \wedge r > y\} \\
(x, y) &= \{r \in R \mid x > r \wedge r > y\}
\end{aligned}$$

3 ROLE-BASED AUTHORIZATION CONSTRAINTS

Constraints are an important aspect of access control and are a powerful mechanism for laying out higher level organizational policy. Consequently the specification of constraints needs to be considered. So far this issue has not received enough attention in the area of role-based access control. [AS00] identified the major classes of constraints in RBAC such as *Prohibition Constraints* and *Obligation Constraints*, including *Cardinality Constraints*. We briefly overview these identified classes of constraints in role-based systems.

3.1 Prohibition Constraints

In organizations, we need to prevent a user from doing (or being) something that he is not allowed to do (or be) based on organizational policy. *Prohibition Constraints* are constraints that forbid the RBAC component from doing (or being) something which it is not allowed to do (or be). A common example of prohibition constraints is SOD. SOD is a fundamental technique for preventing fraud and errors, known and practiced long before the existence of computers. We can consider the following statement as an example of this type of constraint: if a user is assigned to purchasing manager, he cannot be assigned to accounts payable manager. This statement requires that the same individual cannot be assigned to both roles which are declared mutually exclusive.

3.2 Obligation Constraints

We also need to force a user to do (or be) something that he is allowed to do (or be) based on organizational policy. We derived another class of constraints from this motivation. *Obligation Constraints* are constraints that force the RBAC component to do (or be) something. The motivation of this constraints is from the simulation of lattice-based access control in RBAC. There exists a constraint which requires that certain roles should be simultaneously active in the same session. There is another constraint which requires a user to have certain combinations of roles in user-role assignment. We classify this kind of constraints as obligation constraints.

3.3 Cardinality Constraints

Another constraint is a numerical limitation for the number of users, roles, and sessions. For example, only one person can fill the role of department chair; similarly, the number of roles (sessions) an individual user can belong to (activate) could be constrained.

4 CASE STUDY: CONSTRAINTS AND USER-GROUP ASSIGNMENT

Most of role-based constraints work have focused on separation of duty constraints which is a foundational principle in computer security. As a security principle, SOD is used to formulate multi-user control policies, requiring that two or more different users be responsible for the completion of a transaction or set of related transactions. The purpose of this principle is to minimize fraud by spreading the responsibility and authority for an action or task over multiple users, thereby raising the risk involved in committing a fraudulent act by requiring the involvement of more than one individual. A frequently used example is the process of preparing and approving purchase orders. If a single individual prepares and approves purchase orders, it is easy and tempting to prepare and approve a false order and pocket the money. If different users must prepare and approve orders, then committing fraud requires a conspiracy of at least two, which significantly raises the risk of disclosure and capture.

Although separation of duty is easy to motivate and understand intuitively, so far there is no formal basis for expressing this principle in computer security systems. Several definitions of SOD have been given in the literature. We have the following definition for interpreting SOD in role-based environments [AK01].

> **Role-Based separation of duty** *ensures SOD requirements in role-based systems by controlling membership in, activation of, and use of roles as well as permission assignment.*

Separation of duty constraints can be determined by the assignment of individuals to roles at user-assignment time. Consider the case of initiating and authorizing payments. The separation of duty constraints could require that no individual who can serve as payment initiator could also serve as payment authorizer. This could be implemented by ensuring that no one who can perform the initiator role could also be assigned to the authorizer role. This static separation of duty can apply to the user-role assignment. Therefore, we adapt the grant model in URA97. User u can be explicitly assigned to role r_i where $(u$,$r_i) \in UA$. Also user u can be implicitly assigned to role r_j where $(\exists r_i \preceq r_j)[(u, r_j) \in UA]$. Let CR be a set of roles which are needed to be in static SOD. CR is said to be a conflicting role set. The static SOD requirement is that the same user cannot be assigned explicitly or implicitly to more than one role in CR.

We can enforce static SOD as we check each assignment task with a given CR. We have AT-SET (assignment time set) table which includes SOD sets used to enforce SOD requirements at assignment time. The example of AT-SET table with CR is described below. This table tells us that role $pay\_initiator$ and $pay\_authorizer$ are conflicting each other so a user cannot be assigned to both roles.

SET-NAME	ELEMENT
CR$_1$	{ pay_initiator, pay_authorizer }

Whenever System Security Officer (SSO) does assignment tasks, each assignment task should be checked with AT-SET table and satisfy the constraints in the table. Figure 3 describes an algorithm which achieves desired behavior of CONUGA. There are

Grant Algorithm

Let *invoker* be an initiator of user-role assignment and let *assign_DB* have three attributes such as *assign_DB*.`admin`, *assign_DB*.`cond` and *assign_DB*.`range` to construct a table as shown in Table 1.

```
invoker_role_set ← Membership(invoker)
target_role ← role to be assigned
user ← user to which target_role is assigned
assign_DB ← can-assign relation table
CR_set ← AT_SET table
grant_Flag ← false
assign_role_set = $\phi$

While (assign_DB $\neq$ EOF)
    if invoker_role_set exists in assign_DB.admin then
      if target_role exists in assign_DB.range then
        user_role_set ← Membership (user);
        if user_role_set exists in assign_DB.cond then
          grant_Flag = true;
          return;
        endif
      endif
    endif
End

if grant_Flag = true then
    assign_role_set ← JuniorList (target_role);
    if assign_role_set $\cap$ CR_set = $\phi$ then
        do the assignment of role in assign_role_set;
    else
        exit;
    endif
endif
```

Procedure `Membership` (*user*)
Take all assigned roles to a user

Procedure `JuniorList` (*role*)
Take all junior roles to a specified role in role-hierarchies

Fig. 3. Grant Algorithm in CONUGA

two procedures called Membership and JuniorList. Membership procedure allows us to have all assigned roles to a user and JuniorList procedure returns all junior roles to a specified role by walking down the hierarchy. This grant algorithm checks *can-assign* table and AT-SET table to enforce constrained user assignment.

4.1 Implementation Details

Every account in Linux contains a group membership list indicating which groups the account belongs to. Users belonging to a group are explicitly enumerated in either /etc/passwd (for the primary group) or /etc/group (for secondary groups). Many commercial database management systems, such as Informix, Oracle and Sybase, provide facilities for hierarchical groups (or roles). Commercial operating systems, however, provide limited facilities at best for this purpose.

To maintain the group hierarchy we use the file /etc/grouphr to store the children and parents of each group. The group hierarchy of Figure 1 is represented in /etc/grouphr as shown in Table 1. The first column gives the group name, the second column gives the (immediate) parent groups of that group, and the third column gives the (immediate) children. The null symbol "−" means that the group has no parent or child as the case may be. Using /etc/grouphr, we can find all seniors and juniors for a group by respectively chasing the parents and children.

We say a user is an *explicit* member of a group if the user is explicitly designated as a member of the group. A user is an *implicit* member of a group if the user is an explicit member of some senior group. To simulate a group hierarchy we use information about explicit and implicit membership in /etc/group. If Alice belongs explicitly or implicitly to a group she will be added to that group's member list in /etc/group. However, /etc/group is not sufficient to distinguish the case where Alice is both an explicit and implicit member of some group from the case where she is only an implicit member of the group. For this purpose we introduce another file /etc/explicit that keeps information about explicit membership only.

In order to enforce separation of duty constraints, we maintains /etc/at_set which includes conflicting roles. This table also can contain conflicting users and permissions. Table 2 illustrates how we can accommodate such sets to support constrained user assignments.

There are two issues that need to be addressed in decentralized management of group membership. Firstly we would like to control the groups that an administrative group has authority over. Recall figures 1 and 2 which respectively show the regular and administrative groups of our example. We would like to say, for example, that the PSO1 administrative group controls membership in project 1 groups, i.e., E1, PE1, QE1 and PL1. Secondly, it is also important to control which users are eligible for membership in these groups.

URA97 addresses these two issues respectively by means of a *group range* and a *prerequisite group* or more generally a *prerequisite condition*. URA97 has a *can_assign* relation which we store in the file /etc/can_assign. We put a colon between the columns to indicate the boundary. Table 3 illustrates the general case of /etc/can_assign with prerequisite conditions. Let us consider the PSO1 rows. The first row authorizes PSO1 to assign users with prerequisite group ED into E1. The second one authorizes

Table 1. The example group hierarchy of Figure 1

Group Name	Parent Group(s)	Child Group(s)
DIR	-	PL1, PL2
PL1	DIR	PE1, QE1
PL2	DIR	PE2, QE2
PE1	PL1	E1
QE1	PL1	E1
PE2	PL2	E2
QE2	PL2	E2
E1	PE1, QE1	ED
E2	PE2, QE2	ED
ED	E1, E2	E
E	ED	-

Table 2. The example AT-SET table: `/etc/at_set`

Set Name	Elements
conf-roles-1	QE1, QE2
conf-roles-2	PE1, PE2
conf-roles-3	PL1, PL2

Table 3. Example of `/etc/can_assign`with Prerequisite Conditions

Administrative Group	Prerequisite Condition	Group Range
PSO1:	ED :	[E1,E1]:
PSO1:	ED $\wedge$ $\overline{\text{QE1}}$:	[PE1,PE1]:
PSO1:	ED $\wedge$ $\overline{\text{PE1}}$:	[QE1,QE1]:
PSO1:	PE1 $\wedge$ QE1:	[PL1,PL1]:
PSO2:	ED:	[E2,E2]:
PSO2:	ED $\wedge$ $\overline{\text{QE2}}$:	[PE2,PE2]:
PSO2:	ED $\wedge$ $\overline{\text{PE2}}$:	[QE2,QE2]:
PSO2:	PE2 $\wedge$ QE2:	[PL2,PL2]:
DSO:	ED:	(ED,DIR):
SSO:	E:	[ED,ED]:
SSO:	ED:	(ED,DIR]:

PSO1 to assign users satisfying the prerequisite condition that they are members of ED but not members of QE1 to PE1. Taken together the second and third rows authorize PSO1 to put a user who is a member of ED into one but not both of PE1 and QE1. The fourth row authorizes PSO1 to put a user who is a member of both PE1 and QE1 into PL1. Note that, a user could have become a member of both PE1 and QE1 only by actions of a more powerful administrator than PSO1. The rest of table 3 is similarly interpreted.

Table 4. The permission of reference files

PERMISSION	OWNER	Setgid	GROUP	FILE NAME
U:rw- G:rws W:--x	root	YES	rbac	assign
U:rw- G:rws W:--x	root	YES	rbac	weak_revoke
U:rw- G:rws W:--x	root	YES	rbac	strong_revoke
U:rw- G:rw- W:r--	root	NO	rbac	`/etc/group`
U:rw- G:rw- W:r--	root	NO	rbac	`/etc/explicit`
U:rw- G:rw- W:r--	root	NO	rbac	`/etc/can_assign`
U:rw- G:rw- W:r--	root	NO	rbac	`/etc/can_revoke`
U:rw- G:rw- W:r--	root	NO	rbac	`/etc/grouphr`
U:rw- G:rw- W:r--	root	NO	rbac	`/etc/at_set`

Assignment of a user to a group in URA97 means explicit assignment. Implicit assignment to junior groups happens as a consequence and side-effect of explicit assignment. In other words `/etc/can_assign` applies only to explicit membership.

We use the setgid feature of Linux to enforce this behavior. The setgid (set group ID or SGID) file access modes provide a way to grant users access to which they are not otherwise entitled on a temporary, command level basis via a specified program. When a file with SGID access is executed, the effective group ID of the process is changed to the group of the file, acquiring that group's access rights for duration of the program contained in this file. Using setgid a user who is working as an administrative group can read and write the reference files: `/etc/group`, `/etc/explicit`, `/etc/grouphr`, `/etc/can_assign` and `/etc/can_revoke`. Thereby we can enforce desired behavior of URA97 with respect to different administrative groups.

To implement CONUGA in Linux we use several reference files introduced in the previous sections and set their permission bits as shown in table 4. The three procedures assign, weak_revoke and strong_revoke are setgid to the special group `rbac` defined for this purpose. These procedures can read and write the five reference files. We previously described the structure of files `/etc/group`, `/etc/explicit`, `/etc/grouphr`, `/etc/at_set`, `/etc/can_assign` and `/etc/can_revoke`. For simplicity all these files in our implementation are owned by root. We assume that the `rbac` group has no members.

In our implementation a user invokes the procedure call to grant or revoke a group from or to another user. The parameters specify which user is to be assigned to target_group, or to be weakly or strongly revoked from target_group. This implementation is convenient for administrative groups since they only need to define the group hierarchy and the relations `/etc/can_assign` and `/etc/can_revoke`. These procedures are called at the Linux command line prompt as follows.

```
[usage] assign username target_group
[usage] weak_revoke username target_group
[usage] strong_revoke username target_group
```

5 CONCLUSION

In this paper we have described how to extend the Linux group mechanism supporting constrained user group assignment model that is useful in managing group-based access control. When a user is assigned to a group the system checks constraints including prerequisite conditions and conflicting role set, and *automatically* adds the user to all junior groups to the group. We have extended the URA97 model and implemented it in Linux by means of setgid programs. Our result indicates that (static) separation of duty constraints can be determined by the assignment of individuals to groups at user-group assignment time and this behavior can be achieved by accommodating sophisticated access control model to some extent.

References

[AK01] Gail-Joon Ahn and Kwangjo Kim. CONUGA: Constrained User Group Assignment. *Journal of Network and Computer Applications*, 24(2), April 2001.

[AS00] Gail-Joon Ahn and Ravi Sandhu. Role-based authorization constraints specification. *ACM Transactions on Information and System Security*, 3(4):207–226, November 2000.

[FCK95] David Ferraiolo, Janet Cugini, and Richard Kuhn. Role-based access control (RBAC): Features and motivations. In *Proceedings of 11th Annual Computer Security Application Conference*, pages 241–48, New Orleans, LA, December 11-15 1995.

[GI96] Luigi Guiri and Pietro Iglio. A formal model for role-based access control with constraints. In *Proceedings of IEEE Computer Security Foundations Workshop 9*, pages 136–145, Kenmare, Ireland, June 1996.

[JGAS01] James Joshi, Arif Ghafoor, Walid G. Aref, and Eugene H. Spafford. Digital government security infrastructure design challenges. *IEEE Computer*, 34(2):66–72, February 2001.

[NO95] Matunda Nyanchama and Sylvia Osborn. Access rights administration in role-based security systems. In J. Biskup, M. Morgernstern, and C. Landwehr, editors, *Database Security VIII: Status and Prospects*. North-Holland, 1995.

[San97] Ravi Sandhu. Roles versus groups. In *Proceedings of the 1st ACM Workshop on Role-Based Access Control*. ACM, 1997.

[SB97] Ravi Sandhu and Venkata Bhamidipati. The URA97 model for role-based administration of user-role assignment. In T. Y. Lin and Xiaolei Qian, editors, *Database Security XI: Status and Prospects*. North-Holland, 1997.

[SCFY96] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, February 1996.

[YCS95] Charles Youman, Ed Coyne, and Ravi Sandhu, editors. *Proceedings of the 1st ACM Workshop on Role-Based Access Control, Nov 31-Dec. 1, 1995*. ACM, 1995.

Risk Analysis of Biometric Systems

Christos K. Dimitriadis[1], Prof. Despina Polemi[2]

[1] Expertnet SA, 244 Kifisias Av., 15231 Athens, Greece
Christos.Dimitriadis@expertnet.net.gr
[2] University of Piraeus, 80 A. Dimitriou, 18534 Piraeus, Greece
dpolemi@unipi.gr

Abstract: This paper, presents a risk analysis knowledgebase, which aims to enhance existing risk analysis methodologies and tools, by adding the capability of analyzing the risk of the biometric component of an information system. The knowledgebase was created by applying the Multi-Criteria Analysis methodology to the results of research in the security aspect of biometric technologies. The result is a set of vulnerabilities, risk factors and countermeasures for biometric systems.

1 Introduction

A main security weakness of password and token-based authentication mechanisms, is the fact that knowledge, as well as the possession of an item, does not distinguish a person uniquely. Modern biometric technologies provide enhanced security levels by introducing a new dimension in the authentication process called “proof by property”. However, the design and deployment of a security architecture incorporating biometric technologies hides many pitfalls, which when underestimated can lead to major security weaknesses. International security standards, such as ISO/IEC 17799, “IT – Code of practice for information security management” and COBIT: “Control Objectives for Information and related Technology”, provide the general guidelines and principles for correctly deploying a security architecture, both indicating as a very important aspect, the conduct of risk analysis. Regardless of the risk analysis methodology deployed, it is a common practice to utilize a database of common risks and countermeasures (called knowledgebase) for assuring the effectiveness of the process [4]. Such knowledgebases, are sources of expertise regarding security issues for the various components of information systems, assuring that no risks will be missed and adequate countermeasures will be proposed during the process. Despite the existence of biometric-specific standards and best practices, such as ANSI X9.84 “Biometric Information Management and Security” and Best Practices in Testing and Reporting Performance of Biometric Devices [1], there are no detailed knowledgebases for assisting the risk analysis process, as far as biometrics are concerned.

This paper, presents a risk analysis knowledgebase, which aims to enhance existing risk analysis methodologies and tools, by adding the capability of analyzing the risk of the biometric component of the system. Part of this work is the author’s contribution to the EC project BIOSEC [20]. The authors would like to thank the EC for funding BIOSEC, as well as the BIOSEC partners. The remainder of the paper is organized in the following main sections:

- Methodology (general approach and multi-criteria analysis): Describing the methodology followed for building the knowledgebase.
- Biometric risk analysis Knowledgebase (BK): Containing the knowledgebase of vulnerabilities and risk factors of biometric systems, as well as the countermeasures for risk reduction.

2 Methodology: General Approach

In order to ensure applicability and easy integration of BK, to the widest possible risk analysis methodologies and tools, a general risk analysis model [2][3], which is implemented by most of the standard methodologies, was studied. This model is comprised of the following steps:

1. Asset identification,
2. Threat identification (defining as threat, an event that can potentially cause undesirable effects),
3. Vulnerability identification (defining as vulnerability, a security weakness of the system),
4. Risk identification (defining as risk, the probability that a particular threat will exploit a particular vulnerability),
5. Identification of countermeasures for risk reduction.

The first two steps (asset and threat identification) are covered sufficiently by standard risk analysis methodologies without the need of a specialized BK.

The third step (vulnerability identification) revealed the emerging need for the development of a catalogue of vulnerabilities for biometric systems. The catalogue acted as a foundation stone of BK and was populated by conducting:

- Extensive desk research on known or possible attacks against various biometric technologies
- Penetration tests on biometric systems in a dedicated lab
- Interviews with experts in the field

The forth step (risk identification), depends on the risk analysis methodology. Most standard risk analysis methodologies rely on the combination of existing knowledge with information extracted from questionnaires and interviews [5][6]. Other methodologies and tools [7] utilize predetermined risk scores for each identified vulnerability, based on the estimation of experts who studied the likelihood of occurrence of vulnerability exploits. For the creation of BK, a quantitative approach was chosen, calculating a risk factor for each vulnerability. The risk factor is an indicator of the importance of the vulnerability and the sum of all risk factors provides the total risk factor of the biometric component of the information system under review. In order to calculate the risk factor and provide an objective evaluation of each vulnerability a standard methodology called Multi-Criteria Analysis (MCA) was deployed.

The last step (identification of countermeasures for risk reduction), indicated the need for identifying countermeasures for reducing the risk. The countermeasures were identified as an extension of the research conducted for identifying vulnerabilities and was also based on the conduct of tests, desk research and interaction with experts in the field of security and biometrics.

The final form of BK, is a set of vulnerabilities followed by the corresponding risk factors and countermeasures. In the case of vulnerabilities, which are applicable to all biometric technologies, common risk factors were calculated. In the opposite case of technology-specific vulnerabilities, individual risk factors were calculated for fingerprint, iris, face and voice biometrics.

3 Methodology: Application of Multi-criteria Analysis (MCA)

Multi-Criteria Analysis (MCA) [8][18] is a method to evaluate different alternatives (currently biometric vulnerabilities) and to determine an order of ranking of these alternatives. MCA takes into account that some specific criteria should be more influential in the determination of the ranking between alternatives. This is accomplished by the attachment of weighing factors to the different criteria. The following MCA steps were followed for evaluating biometric vulnerabilities:

1. Criteria selection: a number of criteria were selected, which were considered as the most important for evaluating vulnerabilities and which influence their probability of occurrence. These are:

- C1: Difficulty to exploit in terms of technical expertise required and complexity.
- C2: Effectiveness (in terms of level of exposure to threats - binding the vulnerability with the threat).
- C3: Cost in terms of special equipment required.

2. Input of the scores: For each vulnerability a score was calculated per criterion. The score was calculated after processing results from the desk research, biometric lab tests and interviews. The scores were based on a common quantitative scale (from 1-10). In more detail:

- C1: The highest score (10) represents the lowest difficulty.
- C2: The highest score (10) represents the highest effectiveness.
- C3: The highest score (10) represents the lowest cost.

3. Attachment of the weighing factors: The next step in the MCA process involved the prioritization of the criteria by the assignment of different rankings or weights. A weighing factor was attached to each criterion, after studying security incidents and attack profiles. The first three steps of MCA are presented in figure 1:

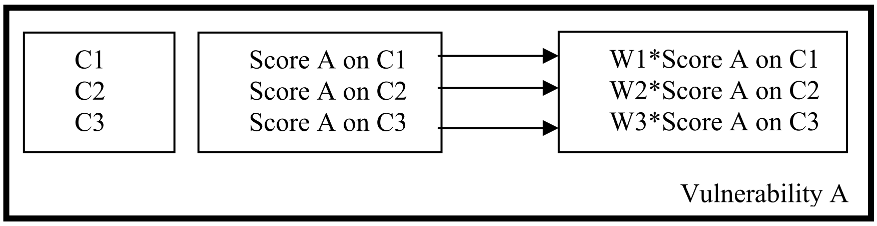


Fig. 1. The first three steps of MCA: criteria *C1*, *C2*, *C3*, *scores* and attached *weighing factors*

4. Ranking of the vulnerabilities: a simple method was deployed - the injunction MCA method. This method multiplies the scores of the criteria with the correspondent weighing factors and calculates for every vulnerability the sum of these products, as shown in the following figure.

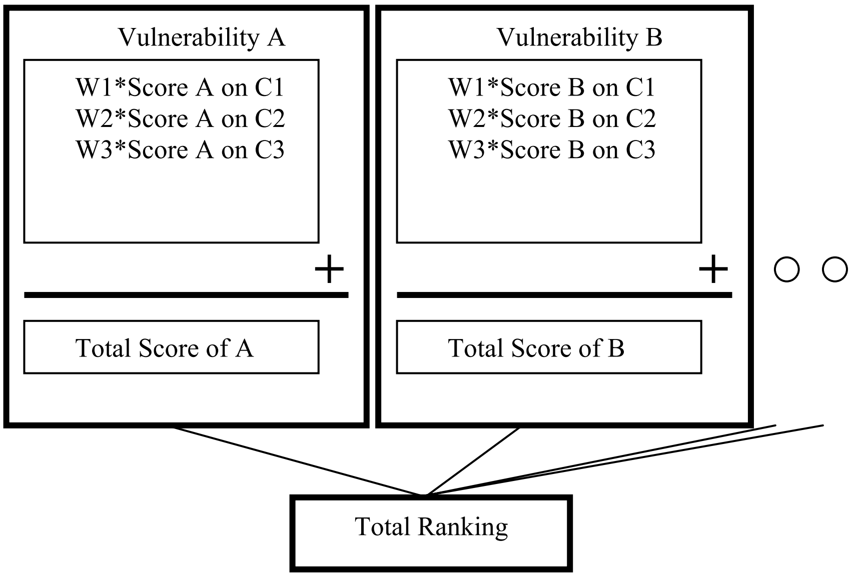


Fig. 2. *Ranking* of *vulnerabilities*. Calculations according to the injunction MCA method

The result is the total score per vulnerability and correspondent ranking. The vulnerability with the highest total score is the highest in ranking and most dangerous.

4 Biometric Risk Analysis Knowledgebase

This section is comprised of two sub-sections. The first one presents the identified vulnerabilities and countermeasures, while the second one presents a comprehensive form of BK, including vulnerabilities, risk factors and countermeasures.

4.1 Description of Vulnerabilities

This sub-section provides a short description of the identified catalogue of vulnerabilities of biometric systems, followed by proposed countermeasures for risk reduction.

- Spoofing – Mimicry – Artefacts: Poor biometric implementations are vulnerable to spoofing and mimicry attacks. An artificial finger made of commercially available silicon or gelatin, can deceive a fingerprint biometric sensor [9][10]. The procedure for materializing this attack is consisted of three steps. The first step is capturing a fingerprint (e.g. from a glass, a door handle or with the user's consent). The second step is creating the artefact, which is a procedure that lasts from a few hours, to a few days maximum. The final step is using the artefact to access the system. The use of pictures, masks, voice recordings or speech synthesis tools is possible to deceive iris, face, and voice recognition systems. As a countermeasure, it must ensured that vitality detection features, which conduct an extra measurement of one or more attributes, such as the relative dielectric constant, the conductivity, the heartbeat, the temperature, the blood pressure, the detection of vitality under the epidermis, or the spontaneous dilation and constriction of the pupil or eye movement, are integrated in the biometric device. If these features are not present, compensating controls must be applied, such as the deployment of multimodal biometrics (e.g. combination of face and lips movement recognition), or the implementation of interactive techniques (e.g. the request for the user to say a specific phrase, or place 3 fingers in a certain order on the sensor).

- Server side - Fake templates: Server based architectures, where the biometric templates are stored centrally, inherit the vulnerabilities of such systems [14]. A possible attack can be realized when the impostor inserts his template in the system under someone else's name. Distributed architectures (e.g. template storage in a smart card) should be preferred. In that case, the template is stored in a tamper resistant memory module that is write-once and erased or destroyed if its content is altered, resisting to this type of attack. When this scenario is not an option, strong security controls must protect the server, including encryption of the templates, system and network security controls (firewalls, intrusion detection and prevention mechanisms) and a strong security policy followed by detailed procedures based on international standards.
- Communication links: Data could be captured from the communication channel, between the various components of a biometric system [14], such as: the sensor and the feature extractor, the feature extractor and the matching algorithm or the matching algorithm and the application, in order to be replayed at another time for gaining access. This is also called electronic impersonation. An effective countermeasure is the integration of the various parts of the system into a hardware security module, or generally the elimination of the transmission of the biometric template. An example of such a module is the biometric smart card, that has a fingerprint sensor and the matching mechanism embedded in it, confining the template to a secure environment. Similar security levels are addressed in integrated terminal devices, such as PDAs or mobile phones. If this is not an option, challenge and response is another approach for addressing this vulnerability. An additional control is the introduction of a rule to discard a signal when it is identical to the stored template or to the last measurement that was conducted.
- Cross system: The utilization of the template in two or more applications with different security levels (i.e. convenience applications and security applications) tends to equalize these security levels, by decreasing the higher security level to the lower one - if a template is compromised in one application, it can be used for gaining access to the other. A countermeasure, depending on the criticality of the application, is the deployment of custom encoding algorithms in order to ensure

the creation of custom templates per user. Another option is the combination of existing biometric encoding algorithms with one-way hash functions for ensuring that the templates produced for a specific user in the specific system, are unique. In that case, special care should be given to the calibration of the system, because very strong non-invertible functions lower the system's accuracy, due to the fact that the matcher, must deal with the measurement variations, in the transformed space [11]. This feature, also provides the ability of revocation to the system in the case that an impostor compromises a template.

- Component alteration: A possible attack can be realized with a Trojan Horse on the feature extractor, the matching algorithm or the decision algorithm of the system, acting as a manipulator of each component's output. Security controls should be defined, such as write-once memory units that host the feature extraction program and the matching algorithm, as well as the integration of the system to a hardware security module. Additional controls include the development of a strong security policy controlling the operation of the system, in order to protect it from exposure to manipulating attempts.
- Enrolment, administration and system use: Poor enrolment, system administration and system use procedures, expose the biometric system. During the enrolment phase, raw biometric data and biometric templates can be compromised and databases can be altered or filled with imprecise user data. Poor system administration procedures, in addition to the above, might lead to altered system configuration files, with decreased FAR, making false acceptance easier, thus security weaker. Similarly, a user might exceed his/her authority, threatening the system. Detailed procedures for user enrolment, system administration and use should be defined, based on international standards and best practices. Controls should be defined, as extensions of the system's security policy, forcing for example segregation of duties, job rotation procedures, logging facilities, alteration or anomaly detection mechanisms.
- Noise and power loss: Off-limit power fluctuation or flooding of a biometric sensor with noise data - for example flashing light on an optical sensor, changing the temperature or humidity of the fingerprint sensor, spraying materials on the surface of a sensor or vibrating the sensor outside its limits - might cause the biometric device to fail. The design of the security policy, should include those security controls that will make the system environment as controlled as possible. These controls depend on the nature of the application.
- Power and timing analysis: Capturing the power consumption of a chip can reveal the software code running on the chip, even the actual command [12][13]. Simple Power Analysis and Differential Power Analysis techniques are deployed for such purposes and are capable for breaking cryptographic algorithms such as DES, by using statistical software. The same strategy can be followed, for breaking the matching mechanism of the biometric system or revealing the biometric template. The secret key or biometric template will appear as the peaks of a diagram projecting the result of applying the appropriate software to the power consumption measurement. Timing attacks are similar and measure the processing time instead of the power consumption. As countermeasure, it should be ensured that all necessary technology controls are in place. These include the use of micro controllers with lower power consumption and noise generators for power blurring. Regarding tim-

ing attacks, the algorithm and program code have to be designed as time-neutral. These technological countermeasures must be included in the biometric system either it is a smart card based architecture or not.

- Residual characteristic: The residual biometric characteristic of a user on the sensor may be sufficient to allow access to an impostor (e.g. a fingerprint the sensor). The attack is realized on a fingerprint sensor with a residual fingerprint from the previous measurement, by pressing a thin plastic bag of warm water on the sensor, by breathing on the sensor or by using dust with graphite, attaching a tape to the dust and pressing the sensor [14]. The last technique is the most effective one. Even when a specific rule in the login algorithm is in place, for declining the exact same measurement, repositioning the tape to provide a slightly different input would deceive the system. A technology assessment should be conducted. Non-optical types of fingerprint sensors are resistant to this vulnerability. In general, deploying interactive authentication in an adequate control for this type of risk.
- Similar template - Similar characteristics: A user having a similar template or a similar characteristic with a legitimate one, might deceive the system, especially in identification applications, where one to many template comparisons are conducted. The maturity of the encoding algorithm, in terms of producing unique outputs from different inputs, as well the FAR of the biometric device should be studied. For security applications the biometric system should be calibrated in order to have reduced FAR (indicative value FAR<0,001%). The maturity of the algorithm can be assured by the deployment of certified products or independently tested products based on [1].
- Brute force: This type of attack is based on trial and error practices [16][17]. The impostor is attempting continuously to enter the system, by sending incrementally increased matching data to the matching function until a successful score is accomplished. This method is most effective in systems that implement identification rather than verification, since the biometric measurement is compared to a great number of templates, making the system weaker (as the number of users increases), due to the increased probability of the existence of similar templates or characteristics among the population. Biometrics however are more resistant to this attack, than traditional systems, since the impostor has to find a way to insert the trial data to the system, thus combine this vulnerability with one of those described above. As a countermeasure, it should be ensured that traditional controls are in place, such as the automatic locking of the user's account after a specific number of attempts, as well as the application of verification instead of identification if possible.

4.2 Comprehensive Form of BK

MCA, was applied step by step for each identified vulnerability. Criteria C1 (difficulty to exploit) and C3 (cost) were assigned with higher weighing factors (equal to 3) than C2 (effectiveness - weighing factor equal to 1), reflecting the most common attack profiles and following the observation that attackers test vulnerability exploits when they are easy to exploit and inexpensive, considering effectiveness at a latter stage [19].

The results of MCA were transformed to percentages (risk factors). Each risk factor indicates the increase of the risk level, in the case that the vulnerability is applicable to the system under review and no countermeasure is taken to address it. The sum of all risk factors provides the total risk factor of the biometric component of the information system under review. The risk factors were individually produced in the cases of vulnerabilities that were specific for each biometric technology. Null scores are translated to non-applicability of the vulnerability to a specific biometric technology. The vulnerabilities, risk factors and countermeasures comprise the comprehensive form of BK(fingerprint: Fi, iris: Ir, face: Fa, voice: V).

Table 1. : Comprehensive form of BK. *Vulnerabilities*, *risk factors* and *countermeasures*

Vulnerability	**Risk Factor (%)**				**CM No.**
	Fi	Ir	Fa	V	
1. Spoofing – mimicry - artefacts	11	10	12	14	i, ii, iii
2. Server side - Fake templates	16				iv, v
3. Communication links	11				Vi
4. Cross system	9				vii
5. Component alteration	11				iv, vi
6. Enrolment, administration and system use	19				iv
7. Noise and power loss	4	4	4	6	iv
8. Power and timing analysis	4				viii
9. Residual characteristic	7	0	0	0	iii, ix
10. Similar template - Similar characteristics	2	2	6	6	ix, x
11. Brute force (verification applications)	4				xi
Countermeasures					
i. Vitality detection.					
ii. Multimodal architecture.					
iii. Interactive authentication.					
iv. Well-implemented security policy according to standards.					
v. Storage of the template in a secure medium.					
vi. System integration into a hardware security module.					
vii. Custom biometric encoding algorithms – hash functions.					
viii. Noise generators, low power consumption chips and specific software design.					
ix. Technology assessment.					
x. Calibration review.					
xi. Traditional controls - account lock after a number of attempts.					

In order to clarify the figures presented in the table, the calculation of the risk factor for the power and timing analysis vulnerability is presented below as an example:

1. Desk research, tests and interviews, defined timing analysis attacks, as difficult to implement (special expertise is required - score on C1=1), effective (score on C2=8) and expensive also (specific equipment is required - score on C3=1).

2. The scores were multiplied with the weighing factor of each criterion, providing a total score of 14.
3. After calculating the total score for each vulnerability, the maximum total score of all vulnerabilities was calculated - it belonged to the case of voice biometrics.
4. All scores were transformed to percentages of the maximum total score of all vulnerabilities. This action was performed, in order to achieve a maximum of 100% when all vulnerabilities are present and at the same time preserve a common denominator for all vulnerabilities. This resulted the risk factor of the power and timing analysis vulnerability to be 4%.

The role of BK during risk analysis depends on the methodology deployed. The main functions are the identification of those vulnerabilities that are applicable to the system under review, after consulting the vulnerability description sub-section, the calculation of the total risk factor, by adding the percentages of each identified vulnerability, utilizing the comprehensive form of BK and the proposal – implementation of countermeasures for risk reduction.

5 Conclusions

The main conclusions of the research conducted, are the following: Special care should be given to user enrolment, system administration and use, implementing as a mandatory control, concrete security policies and procedures based on international standards. Server-based architectures, where templates are stored centrally, heavily increase the risk level of the system, uncovering the demanding need for encryption and strong intrusion prevention, detection and response countermeasures. Vitality detection was also identified as a demanding need, which can be relatively compensated by interactive authentication techniques or multi-modal biometrics. The restriction of the biometric template to a hardware security module and the elimination of the template submission over communication links and networks, addresses a great number of vulnerabilities and reduces the total risk factor significantly. Horizontal results between the four different biometric technologies were also derived and made visible in the comprehensive form of BK, including the high distinctiveness of fingerprint and iris characteristics, reducing the similar characteristic vulnerability. These results however, are strictly related with security, under the specified criteria and should not be confused with results on biometric system performance, or applicability testing. The conduct of risk analysis is a significant step towards the creation of security architectures, which promote the advantages of biometric systems in a risk-proof manner.

References

1. Wayman, J.L., Mansfield, A.J.: Best practices of testing and reporting performance of biometric devices. http://www.cesg.gov.uk/site/ast/biometrics/media/BestPractice.pdf. (2002)
2. Certified Information Systems Auditor Manual. Information Systems Audit and Control Association (2003)

3. Peltier, T.R.: Information Security Risk Analysis. CRC press LLC USA (2001)
4. King, M., Dalton, C., Osmanoglu, T.: Security Architecture. RSA press USA (2001)
5. Operationally Critical Threat, Asset, and Vulnerability Evaluation method (OCTAVE). http://www.cert.org/octave
6. CCTA Risk Analysis and Management Method (CRAMM). http://www.cramm.com.
7. Consultative, Objective and Bi-functional Risk Analysis (COBRA). http://www.security-risk-analysis.com/introcob.htm
8. Multi-Criteria Analysis manual. http://www.odpm.gov.uk
9. Matsumoto, T., Matsumoto, H., Yamada, K., Hoshino, S.: Impact of artificial fingers on fingerprint systems. Proceedings of SPIE, Vol. 4677. Yokohama (2002)
10. Van der Putte, T., Keuning, J.: Biometrical fingerprint recognition – don't get your fingers burned. IFIP TC8/WG8.8 Fourth Working Conference on Smart Card Research and Advanced Applications. Kluwer Academic Publishers. (2000) 289-303
11. Sudan, M., Jules, A.:A fuzzy Vault Scheme. IEEE Internation Symposium on Information Theory. IEEE Press Lausanne Switzerland (2002) 408
12. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic Analysis: Concrete Results. Lecture Notes in Computer Science, Vol. 2162. Springer-Verlag (2001) 251-261
13. Kocher, P., Jaffe, J., Jun, B.: Introduction to Differential Power Analysis and Related Attacks. http://www.cryptography.com/technology/dpa/DPATechnicalInfo.PDF. (1998)
14. IST-1999-20078 Business environment of biometrics involved in e-commerce. http://expertnet.net.gr/bee (2002)
15. Prabhakar, S., Pankanti, S., Jain, A.: Biometric Recognition Security and Privacy Concerns. IEEE Security and Privacy, March /April (2003) 33-42
16. Bolle, R.M., Connell, J.H., Ratha, N.K.: Biometric perils and patches. Pattern Recognition, Vol. 35, no. 12 (2002) 2727-2738
17. Smith, R.: The biometric Dilemma. Secure Computing (2002)
18. Pardalos, P., Siskos, Y., Zopounidis, C.: Advances in Multicriteria Analysis. Kluwer Academic Publishers Dordrecht Hardbound (1995)
19. Know your enemy series. The Honeynet project. http://www.honeynet.org
20. IST-2002-001766 Biometrics and Security – BIOSEC. http://biosec.tid.es

INTEGRATING SECURITY AND PRIVACY ISSUES IN SYSTEM DESIGN

Jan Guynes Clark[1], Nicole Beebe[1] and Andrew G. Kotulic[2]

[1]The University of Texas at San Antonio 6900 N. Loop 1604 West, San Antonio, TX 78258
jgclark@utsa.edu nbeebe@utsa.edu
[2]Kent State University P.O. Box 5190, Kent, OH 44242-0001
akotulic@kent.edu

Abstract. Security and privacy issues are often an afterthought when it comes to system design. However, failure to address these issues during analysis and design could result in catastrophic effects. We propose a conceptual model for creating subsystems of security and privacy that are integral parts of the overall system architecture.

Keywords. Security, Privacy, System Design

1 Introduction

System analysts and designers strive to provide a system that meets the budgetary and business needs of an organization. While they may spend hours tracing the flow of data, few designers pay much attention to the potential security and privacy issues related to the system. We purport that these issues need to be addressed, starting at the earliest stages of analysis and design, progressing through the life of the system. Otherwise, the end result could be a costly, non-aligned system that fails to meet the business needs of the organization. Admittedly, the initial cost of the system would be greater, and the design time would be extended. However, the overall improvement in system efficiency, effectiveness, security, and privacy would be well worth the increased time and effort expended on the design. Additionally, the organization should consider the consequences of not considering security and privacy issues during system design. These could include exorbitant legal costs and civil penalties, along with reduced stakeholder trust.

We propose a conceptual model for system design based upon the integration and interaction of three primary subsystems: business processes, security, and privacy. For this paper, we will focus on the security and privacy subsystems. While no system can maintain maximum privacy and ensure security at all times, this should not prevent us from trying to attain these goals.

Security and privacy goals may seem conflicting and incompatible, especially if they are approached in the later stages of design, or after system implementation. However, if these issues are addressed in the early stages of design, both privacy and security can be attained at a reasonable level.

2 The Systematic Approach

We followed Rechtin's [7] systematic approach to model building: 1) aggregate closely related functions, 2) partition the model into subsystems, and 3) integrate the subsystems into a functioning system. As you will see, there is considerable redundancy in our model. This was intentional. We contend that one's view of a component differs when considering how it relates to the business process, security, and/or privacy subsystem. For example, assume you are designing a patient billing system. While each of the subsystems is concerned with patient data, their view of the data is quite different. The business process subsystem utilizes patient data to charge a given patient for services provided; the security subsystem attempts to prevent patient information from being modified or accessed by unauthorized people; and the privacy subsystem attempts to limit the number of authorized people who can access the data.

We propose that one or more (depending upon project size) members of the design team be assigned responsibility for ensuring compliance with the security and privacy subsystems. Thorough analysis of these subsystems will provide a better understanding of the environment and aid in determining an acceptable level of risk. It will also provide justification for the need for additional expenditures in regard to security and privacy.

Since there is heavy interaction of the components of the system, there should be some degree of overlap among analysts and designers of the subsystems. Additionally, analysis of the components of each of the subsystems should be well documented and stored in a system knowledge database.

Although there is a close relationship between knowledge and data management, they are not the same. Knowledge is frequently fragmented, and signifies the relationships among information, or one's perception or understanding of a given concept. Both are concerned with acquisition and manipulation of data. However, knowledge management focuses on people, culture, and organizational structure, rather than technology.

Knowledge obtained during the system development process should not simply be stored in a database for archival purposes, never to be retrieved. Instead, it should be viewed, updated, and manipulated throughout the lifetime of the system, thus potentially enhancing the success of both current and future system development projects. Lessons learned should be included, because one frequently learns more from failure than success.

Our system framework centers around a shared knowledge base, accessible by everyone who has the need to know. Sharing of information and knowledge enables the analysts and designers to view their given subsystem in light of the other subsystems. This may aid in a better understanding of the system as a whole, and assist in alleviating or mitigating problems from the onset. Building the correct system is not enough. One must also build the system correctly.

2.1 Security Subsystem

The primary focus of the security subsystem (Figure 1) is protection of the organization's information assets. These assets include information and data, software, hardware, people and procedures. In order to provide the appropriate balance between efficiency, effectiveness, security, and privacy of a system, the following components should be addressed:

2.1.1 Security Risk Analysis

The level of security applied to a system, or its components, should be commensurate with the level of assumed risk. Therefore, the system analyst and/or designer must be aware of the potential threats and vulnerabilities associated with the system. Many organizations, such as the Open Web Application Security Project (OWASP) [9] provide information on threats and vulnerabilities, along with steps to be taken to mitigate risks. However, these should be viewed only as guidelines. More pertinent information related to the given system should be obtained from the organization's stakeholders. Once threats and vulnerabilities are determined, one must objectively evaluate the qualitative and/or quantitative impact of a given threat or vulnerabilities. Some threats may seem so remote that they simply are not worth considering, while others may seem imminent. For example, the designers should include password protection on a web-based system that provides access to customer accounts, but not necessarily on one that provides publicly available information. The steps in security risk analysis include the following:

- Identify the system functions, boundaries, and criticalities
- Identify security threats and vulnerabilities
- Evaluate qualitative and/or quantitative impact
- Calculate relative risk factors
- Design cost-effective controls for those threats and vulnerabilities with the greatest relative risk
- Document results of the security risk analysis in the system knowledge database

2.1.2 Data Evaluation

Systems exist in order to manipulate data. Data in some contexts may appear quite innocuous, yet when combined with other data, may be far more revealing. For example, most user ID's are related to an individual's name and can often be determined by simply viewing one's email address. That by itself is not a major security threat. However, a perpetrator could also access the passwords associated with the user ID's of pertinent personnel, potentially resulting in a major threat. Also, data may be considered secure within storage, but how secure is it when it is transmitted from one location to another? Security concerns of the following factors need to be considered:

- Determine the type of each data element within the proposed system – static, dynamic, or derived
- Determine how each data element is to be manipulated – create, store, access, process, transmit, print, and archive

- Classify the data according to access type -Public, Internal, Confidential, Restricted
- Document the data evaluation in the system knowledge database

2.1.3 Security Policies

Steps should be taken to protect data and information assets from unauthorized persons. Clearly defined policies and procedures help to emphasize management's commitment to maintaining security and privacy and instill a more secure culture within an organization. The need for these policies is greatly enhanced in organizations that interact with other entities by way of internetworks. Policy steps include the following:

- Review the security risk analysis to determine its impact on stakeholders
- Review and modify existing security policies, procedures, and documentation based on results of the security risk analysis
- Receive stakeholder approval, where appropriate, of new and/or updated policies, procedures, and documentation
- Distribute the revised policies to the appropriate personnel and stakeholders
- Assure that third parties are aware of the security policies pertaining to the proposed system
- Document security policy changes in the system knowledge database.

2.1.4 Security Legislation and Regulation

System designers must be aware of changes in the legal environment which may impact system requirements. This is always a daunting task, but compounded with organizations that conduct business across national borders. Some regulations, such as the United States' Health Information Portability and Accountability Act (HIPAA) apply only to one country, or group of countries. Others may be more pervasive, such as the Sarbanes-Oxley Act, which applies to all corporations (regardless of physical location) which are publicly traded on the U.S. financial markets [6]. We propose that a team approach be used to monitor the activities of the following bodies in order to deal with the many facets of this problem. The members should come from the security, audit, legal, management, IS/IT and HRM areas, as well as any other functional area, based on the impacted system.

- Review government agencies (Local and Foreign) for changes in security legislation
- Review industry regulatory groups for proposed changes in security practices and legislation
- Review international standards groups, such as the ISO, to assure compliance with the most current and proposed guidelines
- Revise security policies if deemed necessary
- Document changes in the system knowledge database

2.1.5 Security Architecture

As previously stated, security measures are not foolproof. Therefore, overlapping controls should be available to assure an adequate level of protection for the organization's information assets. The existing security architecture and supporting infrastructure should be reviewed and modified, as deemed necessary. A secure architecture requires assessment of every aspect of the system as well the network under which it operates. This includes:

- Review Business Continuity and Disaster Recovery plans
- Review Best Practices of the industry and organization
- Review business and system requirements
- Review physical and environmental protection procedures
- Review physical and system access controls
- Review computer system and application control
- Review information classification, access, and disposal controls
- Review network security infrastructure controls
- Document the changes to the security architecture in the system knowledge database

2.1.6 System Security Integration

System integration is the ability to seamlessly share data and resources across a variety of systems and platforms. Systems security integration takes this one step further by incorporating security into the process. The system designer must ensure that the proposed system security is not negatively impacted by other systems and/or platforms with which it may come in contact. Many organizations have formed strategic alliances which require fully integrated system communication throughout the supply chain. Therefore, the designer must consider the potential security consequences when systems are integrated. The following must be considered:

- Review integration of other systems and platforms within the organization
- Review integration with other systems and platforms external to the organization
- Review potential security risks
- Assess degree of access. Are you providing too much access?
- Assess potential legal and/or ethical ramifications of providing access across multiple platforms and/or organizations
- Establish a record of accountability
- Revise security policies as deemed necessary
- Revise security architecture as deemed necessary
- Document changes in the system knowledge database

2.1.7 Security Training

Policies and controls are of no value if the people expected to abide by them either do not know that they exist, or are not aware of their importance. Approximately 80% of all security breaches occur as a result of user actions (or inactions) that subsequently introduce vulnerabilities into the system [1]. Those who are aware of the consequences of a security breach are more likely to follow safe security practices. Therefore, it is imperative that all potential users be well informed of the importance of maintaining the security of the system, as well as potential consequences of failing to do so.

Security awareness training must be ongoing and should include all levels of the organization, including the top management team. Additionally, partners with whom information from the proposed system will be shared should be required to institute similar programs. The following factors should be considered.

- Provide security-based training to those individuals responsible for creating, storing, accessing, transmitting, printing, and/or archiving sensitive data
- Assure that all legal requirements have been met. For example, select industries such as healthcare and finance are required to provide select security awareness training
- Customize the training, incorporating appropriate policies and procedures
- Document security training (who, what, when, etc.) in the system knowledge database

2.1.8 Knowledge of Security Subsystem

As previously stated the knowledge gained from preparing the security subsystem is to be stored within the system knowledge database. This knowledge can potentially be referenced by system analysts working on the current system, as well as future systems. While some systems may remain relatively static for long periods of time, they are all, to some degree, dynamic. We therefore do not suggest that the knowledge database be your only source of information. Instead, it is to be considered a composite of knowledge regarding data, risk assessments, policies, legislation, training practices, and system architecture and integration over a given period of time.

2.2 Privacy Subsystem

The primary goal of privacy is to ensure the proper handling of personal information, such as one's finances or health status. Organizations can better build trust and customer loyalty if they can show the customers that their personal information is being protected. As with security, total privacy simply cannot be attained unless one lives in total isolation. The primary focus of the privacy subsystem (Figure 1) is to attain an acceptable level of stakeholder privacy. This should ensure that the organization in return merits the level of trust required to conduct its day to day operations with the stakeholder community. In order to provide the appropriate balance between efficiency, effectiveness, security, and privacy of a system, the following components should be addressed:

2.2.1 Privacy Risk Analysis

Potential risks to privacy of the individual and/or organization could arise with the introduction of a new system. Care should be taken in regard to the type of data related to the organization and its stakeholders, and how it is collected, stored, and disseminated. Designers must also consider how manipulation of this data might impact stakeholder perceptions of privacy protection. There appears to be a growing mistrust of consumers toward how organizations protect their personal information. Results in a recent survey showed that consumer confidence in how well businesses handled their personal information dropped from 65% in 1999 to 42% in 2003 [8] .

The steps in privacy risk analysis are the same as those in security risk analysis. However, the focus is on privacy, rather than risk. Those steps include the following:

- Identify privacy threats and vulnerabilities
- Evaluate qualitative and/or quantitative impact
- Calculate relative risk factors
- Design cost-effective controls for those threats and vulnerabilities with the greatest relative risk
- Document results of the privacy risk analysis in the system knowledge database

The analysts and designers should also make note of the following do's and don'ts in an effort to improve the overall system, as well as improve stakeholder trust:

- Provide a means for stakeholders to determine what information is collected about them, and how it is used
- Provide a means for individuals to correct erroneous information about themselves
- Provide a means for individuals to opt in or out of the information collection, processing, or dissemination processes
- Obtain stakeholder consent before disseminating personal data with other organizations
- Do not share personal data with untrusted partners
- Assure the handling of personal data satisfies privacy legislation and abides by the organization's privacy policies
- Review and/or update privacy policies
- Document results of the privacy risk analysis in the system knowledge database

2.2.2 Data Evaluation

Systems that maintain, use, or disseminate individually identifiable information should be designed in a manner to assure confidentiality, integrity, availability, and non-repudiation of the data. The old adage of "garbage in, garbage out" still applies. Data must be obtained from reliable sources, utilizing reliable data collection methods. Control mechanisms also need to be in place to protect against accidental or unauthorized data manipulation. Analysts and/or designers will evaluate the same data characteristics as described in the security subsystem, but their focus will be on data privacy, rather than data security:

- Determine the type of each data element within the proposed system – static, dynamic, or derived
- Determine how each data element is to be manipulated – create, store, access, process, transmit, print, and archive

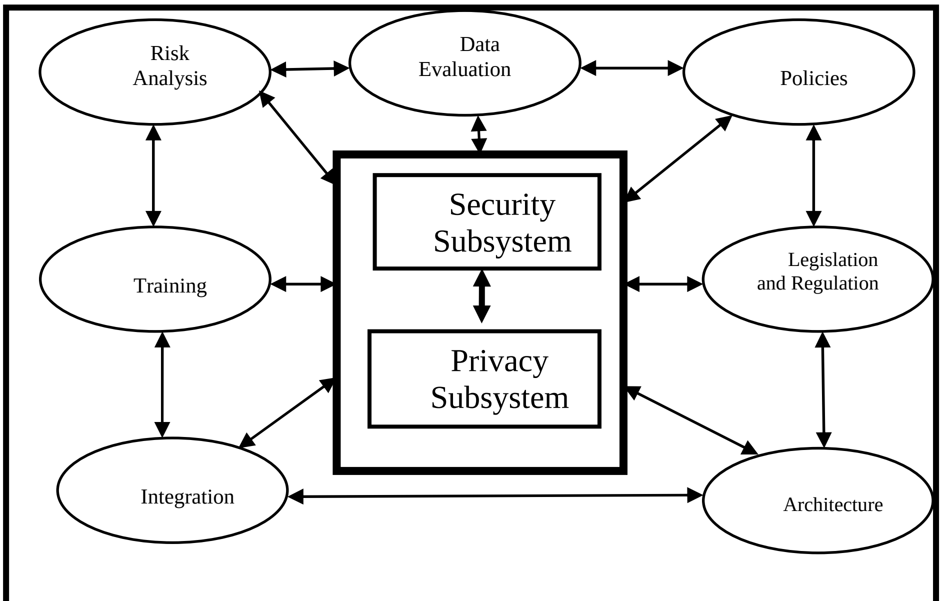


Figure 1: Security and Privacy Subsystems

- Classify the data according access type -Public, Internal, Confidential, Restricted
- Ensure proper protection and treatment of all personally identifiable data. Classify according to risk, value, ownership, and flow within the proposed system [4]
- Establish an audit trail
- Restrict information flow, when possible, when the risk of privacy loss is great
- Document the data evaluation in the system knowledge database

2.2.3 Privacy Policies

There is increasing privacy concern of internetworked systems. We have experienced an exponential rise in invasive software employed by third parties to collect user keystrokes and track their movement throughout the Internet [2]. While many marketers view this as a legitimate way of conducting business, most consumers consider this a violation of their privacy. Analysts and designers must be aware of these potential privacy invasions and take steps to mitigate them. Additionally, the designer must review the organization's privacy policies and design the system accordingly.

- Ensure the existence of a privacy policy that includes clear delineation and agreement with expectation of privacy "rights"

- Determine ownership and responsibility for the policy
- Review the privacy risk analysis to determine its impact on stakeholders
- Review and modify existing privacy policies, procedures, and documentation based on results of the privacy risk analysis
- Receive stakeholder approval, where appropriate, of new and/or updated policies, procedures, and documentation,
- Distribute the revised policies to the appropriate personnel and stakeholders
- Assure that third parties are aware of the privacy policies pertaining to the proposed system
- Document privacy policy changes in the system knowledge database.

2.2.4 Privacy Legislation and Regulation

As with security issues, system designers must be aware of changes in the legal environment that may impact how privacy issues should be considered when designing systems. Customers are becoming increasingly concerned about the data collected about them and how this data is disseminated. The EU is far advanced in preserving the privacy of the individual, while the United States is just beginning to address this issue. Regulations such as HIPAA and the Gramm-Leach-Bliley Act (protection of financial data) are helping to close the gap between the United States and the EU in this regard [5]. Again, we propose that a team approach be used to monitor the activities of the following bodies in order to deal with the many facets of this problem. The members should come from the security, privacy, audit, legal, management, IS/IT and HRM areas, as well as any other functional area, based on the impacted system.

- Review government agencies (Local and Foreign) for changes in privacy legislation
- Review industry regulatory groups for proposed changes in privacy practices and legislation
- Review international standards groups, such as the ISO, to assure compliance with the most current and proposed guidelines
- Ensure compliance with regulations by reviewing procedures for conducting privacy audits, reporting sensitive data, and handling breaches in privacy
- Determine the data to be protected – where is it? Who controls it? How is it accessed?
- Determine the consequences of breaches in privacy – how was it breached? How, and to whom, should the breach be reported? How can we prevent this occurring again?
- Revise privacy policies if deemed necessary
- Document changes in the system knowledge database

2.2.5 Privacy Architecture

The privacy architecture attempts to address privacy concerns as they arise and find ways to introduce privacy-enhancing components into the system architecture. At a recent Privacy Enhancing Technologies (PETs) workshop, the participants concluded that PETs should 1) provide the highest degree of anonymity possible, 2) minimize the amount of data collected about an individual, 3) focus on systems and infrastructures as well as tools, and 4) be designed within a system, rather than added at a later date [3]. Steps include the following:

- Determine privacy requirements
- Formulate potential solutions to the requirements
- Select the best solution, based upon needs of your organization and the data involved
- Integrate the solution with the system design criteria
- Document changes to the privacy architecture in the system knowledge database

The designer must ensure that the technologies being incorporated into the system do not violate the existing internal and external privacy policies. All too often, designers either assume that stakeholder privacy is not compromised, or simply are not aware of its importance. One should pay particular attention to such technologies and procedures as web server log files, cookies, known software bugs and patches, and sophisticated data mining algorithms.

2.2.6 System Privacy Integration

We define System Privacy Integration as the ability to seamlessly share data and resources across a variety of systems and platforms while concurrently protecting stakeholder and corporate privacy.

It is important that organizations routinely monitor and/or evaluate their privacy practices, as well as those of their business partners. The following must be considered:

- Review integration of other systems and platforms within the organization
- Review integration with other systems and platforms external to the organization
- Review potential privacy risks
- Revise privacy policies as deemed necessary
- Revise privacy architecture as deemed necessary
- Document changes in the system knowledge database

2.2.7 Privacy Training

Individuals must understand how to protect the privacy of data. They must also understand the consequences of what could happen when privacy has been breached. Individuals that interact with the proposed system must be aware of all pertinent privacy policies and be expected to abide by them. The need for privacy awareness training must be ongoing and should include all levels of the organization, as well as partners with whom the system information will be shared. The same factors considered for security should be considered for privacy.

- Provide privacy-based training to those individuals responsible for creating, storing, accessing, transmitting, printing, and/or archiving sensitive personal data
- Customize the privacy awareness training, incorporating appropriate regulations, policies, and procedures

- Document privacy training (who, what, when, etc.) in the system knowledge database

2.2.7 Knowledge of Privacy Subsystem

Please note that the system knowledge database may contain a lot of data pertaining to stakeholder privacy and organizational business practices. Therefore, it should be well protected from potential misuse. Only those with the need to know should be provided access to the knowledge database.

3 System Integration/Optimization

One must bear in mind that the privacy, security, and business process subsystems must be fully integrated (Fig. 2). This is a highly iterative process. A change in any of the components in any of the given subsystems requires review of all other components within the system in order to assure efficiency, security, and privacy of the system as a whole. This need for system integration further highlights the necessity of having an updated system knowledge database.

System design knowledge has traditionally been managed via system design documents and configuration management (CM) systems and processes. Such mechanisms, however, seldom document information protection objectives and matrix subsystem design components to those objectives. Traditional configuration management mechanisms primarily serve as inventory management aids, as well as organizational tools in software development environments.

Conversely, the systems knowledge database is intended to be a decision support tool. It helps analysts and developers who have different and sometimes contradictory information protection goals to make sound subsystem design decisions by considering the overarching information protection goals and the impact of changes on other subsystems.

A systems design knowledge database should store security and privacy objectives; results of the risk analysis, including asset identification and valuation, threats and vulnerabilities, and risk management decisions; and resultant subsystem design components implemented. Each design component should be mapped to a set of technical capabilities, as well as the overarching information protection goal(s) addressed by each component. This will facilitate better decision making in later design reviews. When new components are being proposed and legacy components are being considered for removal from the system design, the system knowledge database can be polled and assist in providing detailed information regarding the impact of such system design additions and deletions.

While optimizing the subsystems, the analysts and designers may note conflicts among the subsystems. Complying with one set of regulations or demands may result in the unintentional violation of others. Some conflicts can be addressed without negatively impacting system efficiency, security, and/or privacy, while others may not. As a result, trade-off decisions must be made, and one or more of the subsystems may have to be sub-optimized. Which is more important – security, privacy, or efficiency of the business process? There is no easy answer as to this question. We must be able to efficiently and securely collect, process, and store data while protecting the privacy rights of an identifiable entity. This dilemma is further compounded when one considers where this data is located, and where it may be disseminated. If it crosses national borders, the privacy and security concerns and regulations of each of the involved countries must be addressed.

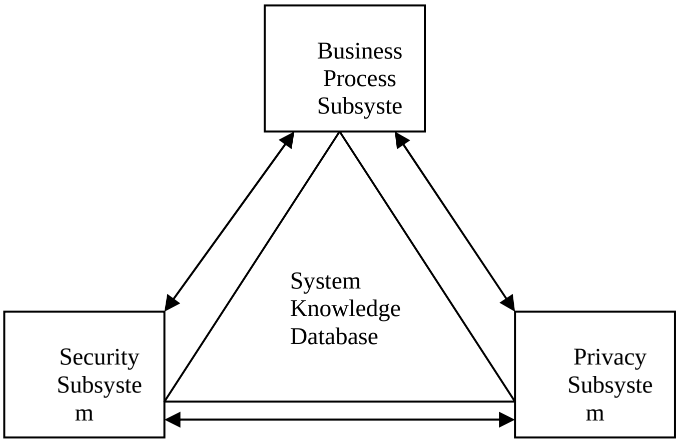


Figure 2: Fully Integrated System

Given a choice, most organizations would probably prefer to compromise the privacy subsystem. Why? Privacy generally impacts its stakeholders, rather than the organization itself; increased privacy controls can, and often do, impact system efficiency; and it is costly and time-consuming to protect stakeholder privacy. However, the organization runs the risk of losing stakeholder trust, which could have a very strong negative impact on the viability of the firm.

Although there is no easy answer to this question, we suggest the following:

- Consider the stakeholders and their level of involvement in the given system
- Identify stakeholder data which has security and/or privacy characteristics (i.e. patient medical records)
- Identify locations internal and external to the organization in which this data could be disseminated
- Perform security and privacy risk analyses
- Evaluate current regulations, policies, and best practices as they relate to the co-located data
- Categorize security, privacy, and business risks
- Address the risks in each category which can be mitigated in a costly manner
- Continually monitor the system throughout the life cycle for changes in security, privacy, and business process

4 Conclusions

As shown in Figure 1, while the components of the security and privacy subsystems are identical, the focus on these components is quite different. While we can never achieve maximum system efficiency within a totally secure and private environment, we can attempt to improve each of these subsystems by addressing them from the onset of system design.

This should lead to a functional system that has taken into consideration the following:
Concerns for security and privacy should not be considered a necessary evil; instead, they should be incorporated within the organizational culture, and viewed as arequirement for maintaining viability of the organization

- Although security and privacy breaches are inevitable, we must strive to reduce them and mitigate consequences of those that occur
- One's employees remain the greatest security risk. Most security violations are unintententional, while others are the result of disgruntled employees. Therefore, organizations should assure their employees are well trained and satisfied with their jobs.
- Security is everyone's responsibility – from the CEO to the first line employee
- The optimal level of security for an organization should be based upon the evaluation of the costs related to obtaining an acceptable risk level
- The major tradeoffs between cost, flexibility, and ease of use should be considered when designing the overall system.

Security and privacy are shared responsibilities. By integrating these susbystems with the business process during the early stages of system design, and by following the basic guidelines, the resulting system should be far more secure, effective, and trustworthy.

References

1. Cowens, B.: The Security Threat Inside: Building an Awareness Program and Effectively Training Your Staff. The ISSA Journal. November 2003. 10-12.
2. Lawton, G.: Invasive Software: Who's Inside Your Computer? Computer. July 2002. 15-18.
3. Main Outcomes of the Technical Workshop on Privacy-Enhancing Technologies 4 July 2003. http://europa.eu.int/comm/internal_market/privacy/docs/lawreport/pet/200304-pet-outcome_en.pdf
4. Mathur, Sanjay.: Builing an Inside-Out Privacy Compliance Framework. The ISSA Journal. December 2003. 14-17.
5. Noor, A.: Dealing with data Privacy Regulations and SB-1386. The ISSA Journal. May 2003. 8-10.
6. Raval, V. Guidelines for Compliance with Sarbanes-Oxley. EDPACS . January 2004. 14-20.
7. Rechtin, E.: Systems Architecting: Creating and Building Complex Systems, Prentice Hall, New York 1991.

8. Taylor, Humphrey. Most People Are “Privacy Pragmatists” Who, While Concerned About Privacy, will Sometimes Trade it Off for Other Benefits. The Harris Poll # 17, March 19, 2003. Http://www.harrisinteractive.com/harris_poll/.
9. The Open Web Application Security Project (OWASP). January 13, 2003. The Ten Most Critical Web Application Security Vulnerabilities. http://www.owasp.org.

Intrusion Risk Analysis and the Power Law Distribution of Attacks

Juan Manuel Garcia Garcia

Department of Computer Systems
Instituto Tecnológico de Morelia
Morelia, Mexico
E-mail: jmgarcia@sekureit.com

Abstract. Risk analysis is the first essential step in the risk management process. In order to do an effective risk analysis, is necessary to identify and quantify the threats to information technology assets. Then statistical models of information security threats are required to develop effective risk analysis methodologies. We present experimental evidence suggesting that network intrusion attacks follows a power law distribution and then we explore some implications for intrusion risk analysis.

Keywords: Risk analysis, network intrusion detection, power law distribution.

1 Introduction

Risk analysis, also known as risk assessment, is the first essential step in the information security architecture deployment [11]. Information security controls implemented in any organization should be commensurate with its risks. The purpose of information security risk analysis is to determine, on the most objective basis, which security controls are appropriate and cost effective.

There are several approaches to risk analysis. However, these can be classified into two categories: quantitative and qualitative. [10]

In the quantitative approach, probability data is not required and only estimated potential loss is used. The main drawback of this approach is its very subjective nature and that it heavily relies on the expertise of the risk analysis team's members. However, it is the most widely used approach to risk analysis.

The quantitative approach focus on two elements: the probability of an event occurring and the likely loss should it happen. Quantitative risk analysis make use of a single indicator called the *annual loss expectancy* (ALE), calculated for an event as the product of the potential loss by the probability of the event occurrence. Then it is possible to rank events in order of risk and to make decisions about control and countermeasures based on this. The effectiveness of this approach depends on the reliability and accuracy of the statistical data associated with the event.

One of the most important security controls to be considered into an information security architecture are *intrusion detection systems, (IDS)* [2][8]. An IDS needs to be cost-effective in the sense that it should cost no more than the expected level of loss from intrusions. Despite this, IDS cost-benefit analysis is seldom done.

In a previous work [6], major cost factor associated with an IDS were examined, including development cost, operational cost, damage cost due to successful intrusions, and the cost of manual and automated response to intrusions. Cost factors are qualified according to a definite attack taxonomy [7], which main categories are illegal root access, illegal user access, denial of service, and information gathering. A cost-benefit analysis methodology for network intrusion detection had been proposed [12], based on an investigation of the cost factors and categories of various intrusions.

But, in order to do a full cost-benefit analysis for NIDS, a loss expectancy value must be obtained, and then a model for potential loss and probability of intrusions is required. In this paper, we present a probabilistic model for network intrusions that follows a power law distribution.

2 Experimental Data on Attacks

In order to get some insight about what kind of probabilistic distribution follows network intrusion events we analyse data collected by an IDS over several months. The IDS used was Snort 1.8 [1] using arachNIDS database[2] of network attacks signatures.

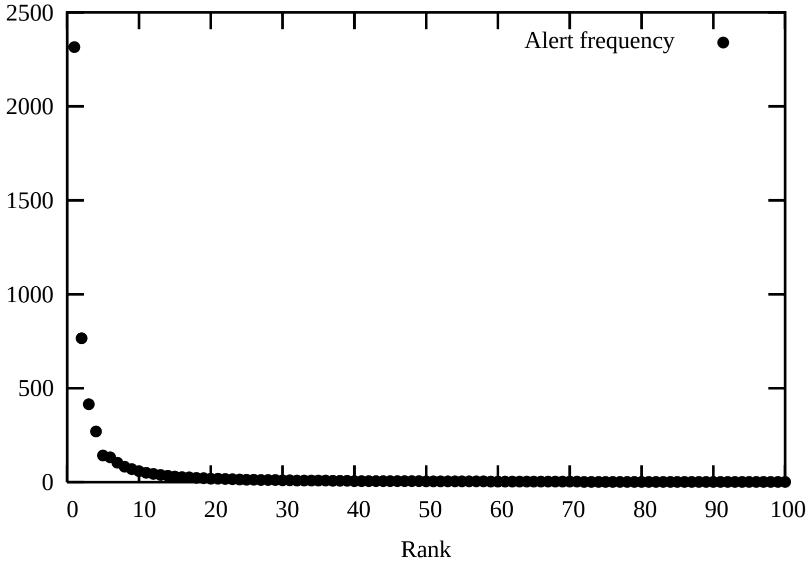


Fig. 1. Frequency versus rank of alerts.

Over eight months of observation, 5091 alerts were logged, 137 different kind of attacks were observed, where arachNIDS database includes more than 500 different attack signatures. The most frequent kind of attack logged was port scanning, like SYN FIN Scan (IDS198) or Probe SYN Scan (IDS441), and one of the most rare attacks logged was, for example, a buffer overflow attempt on IMAP service (IDS147) with a single occurrence on the observed period. Virus and worms attacks were not considered because they are not longer supported by Snort.

After ranking attacks by frequency of occurrence, we observed that the most innocuous attacks, those included in the information gathering category, are also the most frequently seen. We

[1] http://www.snort.org/

[2] http://www.whitehats.com/ids/

observed a high incidence of few probe kind of attack and a very low incidence of more varied and more dangerous attacks, like buffer overflow attempts. In figure 1 we show a plot of rank versus frequency, when can be observed how few attacks are most frequently seen. This observation suggest a power law distribution like that related to many phenomena like Internet traffic [4], web requests [3], etc.

When we plot rank against frequency in a log-log scale we obtain what is showed in figure 2. The least-squares regression line is

$$\log_2 f = -1.62 \cdot \log_2 r + 11.24 \tag{1}$$

where f is the alert frequency, and r is the alert rank. Then we have

$$f = 2418.67 \cdot r^{-1.62}. \tag{2}$$

Then we can take as *cumulative distribution function*

$$F(r) \sim r^{-1.62} \tag{3}$$

Then we can deduce for intrusion attacks the following probability distribution function

$$P(r) = \Omega r^{-\alpha} \tag{4}$$

where $\Omega = 0.46296$ and $\alpha = 1 + 1/1.62 = 1.61728395$. The previous PDF is valid only for $1 \leq r \leq 137$. This is the well known power law or Zipf-like distribution [1]. For an unknown number of attacks, we can generalize the attack distribution to a *zeta distribution*

$$P(r) = \frac{1}{\zeta(\alpha)} \cdot r^{-\alpha} \tag{5}$$

where $\zeta(\alpha)$ is the *Riemann's zeta function* evaluated on α [5].

3 Implications for Intrusion Risk Analysis

In this section, we explore some implications of the power law distribution of attacks that we already present, for intrusion risk analysis. Let $l(r)$ be the potential loss caused by the occurrence of the rank r attack, then the *expected loss* for a set R of attacks would be

$$L = \Omega \sum_{r \in R} l(r) \cdot r^{-\alpha}. \tag{6}$$

Also we can have a *cost* $C(r)$ for detection and response to an event of rank r, that can be calculated by the methodology proposed in [6] and [12].

For a fixed budget B, we want to detect and respond to the attacks that could inflict the greater loss. Then we can formulate intrusion risk analysis as a combinatorial optimization problem in the following way: To find a set of possible threats R such that L, as defined in (6), is maximized, subject to the constraint

$$B = \sum_{r \in R} C(r). \tag{7}$$

In the general case, this problem could be hard to solve, but under some assumptions it could be simplified.

First of all, expected loss (6) can be estimated by its continuous limit

$$L = \frac{1}{\alpha - 1} \int_1^{\infty} l(r) \cdot r^{-\alpha} dr \tag{8}$$

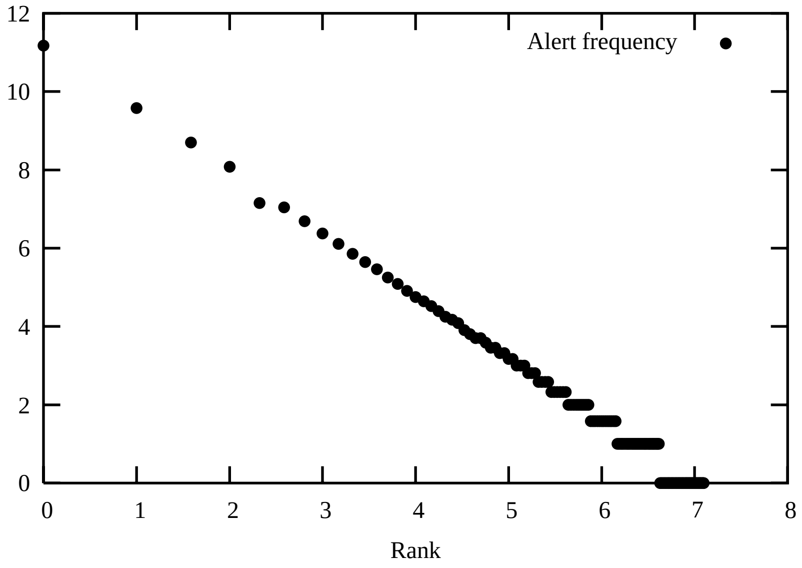


Fig. 2. Frequency versus rank of alerts in a log-log scale.

We can observe that this integral can be computed as the *Mellin transform* [9] of the loss function:

$$\mathcal{M}[l(t)] = \int_0^\infty t^{z-1} \cdot l(t)dt \tag{9}$$

evaluated at $z = 1 - \alpha$.

A simple closed solution can be found under some assumptions. These assumptions are the following:

1. *The potential loss caused by an attack depends only on the kind of attack.* That is, the damage inflicted by, for example, a denial of service, is the same doesn't matter what particular DoS attack it is. (For this matter, we can use the attack taxonomy presented in [7].)
2. *Similar kind of attacks are close ranked.* That is, all the attacks of the same category (see [7]) are ranked in some rank interval $[a, b]$, and all the events ranked in this interval are of the same kind.

These two conditions are very likely from what we have experimentally observed.

If all attacks of the same kind are ranked in some interval $[a, b]$ and the loss caused for any of them is a constant λ then the loss function for that kind of attack can be expressed as the *boxcar function*:

$$B_\lambda(a, b) = \lambda[H(r - a) - H(r - b)] \tag{10}$$

which is equal to λ for $a \leq r \leq b$ and 0 otherwise, and where H is the Heaviside step function. The Mellin transform for this function is

$$\mathcal{M}[B_\lambda(a, b)] = -\frac{\lambda}{z}\left[a^z - b^z\right] \tag{11}$$

and then, the expected loss for that particular kind of attack would be

$$\bar{B}_\lambda(a, b) = \frac{\lambda}{(\alpha - 1)^2}[a^{1-\alpha} - b^{1-\alpha}] \tag{12}$$

Under assumptions previously stated, attacks can be classified into n different kinds, so that for all the attacks of kind k, the associated loss to any of them is a constant λ_k and all of them rank between an interval $[a_k, b_k]$ where a_k is the most probable attack of kind k and b_k the least. Then, we have the *expected loss for attack kind k* as

$$L_k = \frac{\lambda_k}{(\alpha - 1)^2}[a_k^{1-\alpha} - b_k^{1-\alpha}] \tag{13}$$

for $k = 1, \ldots, n$, where the total *expected loss* would be

$$L = \sum_{k=1}^{n} L_k. \tag{14}$$

Estimated costs for attack prevention, detection and response are much more easier to obtain for whole attack kinds than for particular attacks [7]. Thus, if we define C_k as the cost associated to prevention, detection and response to the k attack kind, where $k = 1, \ldots, n$, then cost/benefit analysis can be obtained from L_k and C_k values using a methodology like the exposed in [12].

4 Conclusions and Future Work

We have presented some experimental evidence suggesting that intruders attacks follows a power law distribution, very similar to the kind of distribution associated to several aspects of Internet traffic.

We have shown how this power law distribution can be used to estimate expected losses for diferent kind of attacks, assuming that the loss inflicted by one attack depends only on the kind of attack and that attacks of the same kind are close-ranked. Further experimental evidence is needed to verify how valid are these assumptions.

Further experimental study is also required to extend our analysis to virus and worms attacks.

References

1. R. J. Adler, R. E. Feldman and M. S. Taqqu (eds). *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, Birkhauser, Boston, 1998.
2. R. G. Bace. *Intrusion Detection*, QUE, 1st Edition, December 1999.
3. L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker. Web caching and Zipf-like distributions: evidence and implications. *Proceedings of INFOCOMM'99*, IEEE Press, 2000.
4. A. B. Downey. Evidence for long-tailed distributions in the Internet. *ACM SIGCOMM Internet Measurement Workshop*, November 2001.
5. H. M. Edwards. *Riemann's Zeta Function*. Dover Pubns, June 2001.
6. W. Lee, W. Fan, M. Miller, S.J. Stolfo and E. Zadok. Toward Cost-Sensitive Modeling for Intrusion Detection and Response. *Workshop on Intrusion Detection and Prevention, 7th ACM Conference on Computer Security*, Athens, November 2000.
7. U. Lindqvist and E. Jonsson. How to systematically classify computer security intrusions. *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland CA, May 1997.
8. S. Northcutt and J. Novak. *Network Intrusion Detection*, QUE, 3rd Edition, August 2002.
9. R.S. Pathak. *Integral Transforms of Generalized Functions and Their Applications*. Taylor & Francis, December 1997.
10. T.R. Peltier. *Information Security Risk Analysis*, Auerbach Pub., 1st. edition, January 2001.

11. J.K. Tudor. *Information Security Architecture: An Integrated Approach to Security in the Organization*, CRC Press, September 2000.
12. H. Wei, D. Frinke, O. Carter and C. Ritter. Cost-Benefit Analysis for Network Intrusion Detection Systems, *CSI 28th Annual Computer Security Conference*, Washington D.C., October 2001.

An Efficient Off-Line Reputation Scheme Using Articulated Certificates

Roslan Ismail[1], Colin Boyd[1], Audun Josang[2], and Selwyn Russell[1]

[1] ISRC, Queensland University of Technology, Brisbane, Australia
[2] DSTC, University of Queensland, St Lucia, Australia

Abstract. A common feature practical reputation schemes is that they are on-line which results in restrictions to both availability and scalability. In order to overcome these two problems we propose an off-line reputation scheme based on public key certificates. We introduce the idea of *articulated certificates* which use proxy signatures to increase the efficiency of reputation verification. As well as being well-suited to our problem such linked certificates may be of independent interest.

1 Introduction

Reputation schemes are systems developed to collect, analyse and propagate users' reputation [12]. They can be used for many purposes, but in the last few years have emerged as a promising means for enabling electronic transactions in e-commerce. Studies have shown that use of reputation schemes has positive effects on the efficiency and honesty of markets [1], and that the reputation of a particular agent can have positive effects on the agent's gain [13].

Most current reputation schemes, especially the practical ones, are specifically designed for on-line use, eBay being a prime example (`www.ebay.com`). This situation is not surprising as a reputation scheme must provide real time responses so that users' past behaviours can be obtained immediately. Although an on-line reputation scheme is currently preferred, it suffers two main difficulties: availability and scalability. For example, in a case of denial of service it may not be possible to access the central server and so reputation values cannot be found. If the central server is distributed to overcome such problems, then synchronisation and consistency of data will become difficult.

These problems of distributed reputation systems have much in common with the problems of distribution of public keys. In both cases there is a need for access to authenticated values distributed in a timely fashion. Public keys are usually propagated through *certificates* formed by an off-line trusted third party. It seems a natural idea to use *reputation certificates* formed in an analogous way by a trusted third party. One of the main aims of this paper is to explore how this may best be achieved.

Reputation certificates may be controlled by users themselves. The certified reputation value calculated from processed feedback is communicated to the reputation owner after completion of transactions with its counterparts. Reputation

Table 1. The participants and their symbols

FT	A feedback target is the entity who is being evaluated and gains the reputation rating based on the feedback given by a feedback provider.
RP	A relying party is the entity who relies on the feedback target's reputation rating to make a decision whether to proceed in a transaction or not.
CA	The certificate authority is responsible for the registration of the feedback targets as well as to issue certificates to them.
CC/RA	The collection centre/reputation authority collects legitimate feedbacks and uses them to calculate reputation rating and update the feedback target's reputation certificate.
AA	The attribute authority is responsible for issuing and signing the attributes.

certificates can be obtained from reputation owners without the need to contact a central authority. There seem to be two natural ways to realize this proposal.

1. Employ existing identity certificate technologies, for example, X.509 [5] and PGP [15] to incorporate reputation values.
2. Employ a separate certificate specifically for the reputation value.

In the former option a reputation rating is regarded as one of the attributes in the identity certificate. As a result the implementation does not require any significant modification to the existing infrastructure. The latter, on the other hand, requires a special authority to manage the reputation rating scheme. We will compare the relative advantages of these different options later.

This paper proposes an off-line reputation scheme based on public key certificates. The solution is flexible enough to accommodate most formats of reputation rating. Different options are considered for how to bind the reputation information with the identity of the subject. Our proposal, which we call *articulated certificates*, can be applied in other situations when it is desired to augment or update certificate information without re-issuing the identity certificate.

Organisation of the paper Section 2 discusses the background of reputation schemes. Section 3 discusses three basic solutions to implement binding between identity certificates and reputation information. Section 4 describes our proposed solution, its properties and the required protocols. Section 5 discusses the relative merits with other options. Table 1 presents the notations and the symbols used throughout the paper.

2 Reputation Systems

There has been considerable interest in reputation systems in recent years and an extensive literature has developed [12]. Reputation systems may be roughly classified into two activities.

Reputation calculation is the task of obtaining reputation values from a set of feedback information. There are various properties that may be desirable for calculation engines and reputation values may take different formats. In this paper we are not concerned with how reputation values are calculated, as long as they can be represented efficiently in a bit string.

Reputation propagation is concerned with how to distribute reputation values to parties that require to use them. This is the area addressed in this paper. There are different properties that may be important, including high availability of values and reliability. A feature that has often been neglected is privacy of reputation values; we address this partially in this paper by allowing owners of reputation certificates to control their distribution.

Off-line reputation propagation has been proposed by some recent authors [4, 3]. These schemes addressed the integrity of the submitted feedbacks against manipulation but are not suited to centralised reputation calculation. A recent proposal of Liau et al. [7] (the LZBT scheme) demonstrated the possibility of using certificates to represent a user's reputation in the off-line environment. The LZBT scheme seems promising for P2P systems because no central authority is required to operate the scheme. However, its major limitation is that the relying party has to contact one or more of the preceding feedback provider to verify the validity of reputation certificates. This creates an extra burden to the service consumers to verify the certificate.

3 Reputation certificates

Identity certificates bind the identities of users with their public keys. The certificate is issued and signed by a trusted *certificate authority* CA. Identity certificates are typically long term and contain several attributes such as subject name, public key, expiry date, issuer name, and certificate holder's name. These certificates are mainly used for authentication purposes. Attribute certificates [9], on the other hand, are mainly used to provide access controls and role permissions of an entity with regard to accessing resources. Therefore, these certificates are often employed within organizational boundaries. Attribute certificates typically do not contain the identity of an entity; instead they may contain attributes such as role, access control, expiry date, the issuer name and the issuer signature.

A reputation rating can be considered as an attribute bound to an identity. Reputation certificates therefore need to be used in conjunction with an identity certificate. There are various ways that this may be achieved. Park and Sandhu [11] discussed three techniques to bind two certificates (the identity and the attribute certificate): monolithic, autonomic and chained signatures. In the *monolithic* signature technique the identity and attribute certificate are combined to become a single certificate which is signed by an authority. The *autonomic* signature technique implements separate signatures: the identity certificate and the attribute certificate are signed by different authorities. To bind the two certificates certain attributes in the identity certificate are linked to the attribute certificate. Finally in the *chained signature* technique the signature of authority on the identity certificate is used as a connection link between the identity and the attribute certificates. In the next subsections we will consider solutions which correspond roughly to this classification.

3.1 Combined Certificates

In this solution the identity certificate and reputation certificate are the same object, and the reputation value is simply an additional attribute in the certificate. This corresponds to the monolithic certificate of Park and Sandhu [11]. Figure 1 depicts the abstract view of the solution. In this solution, the feedback target and the feedback provider are required to register with the authority. A reputation certificate is issued and signed by the reputation certificate authority. The reputation certificate should be verified by the relying party.

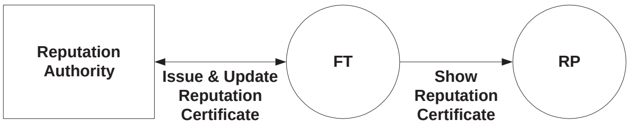

Reputation Authority
Issue & Update Reputation Certificate
FT
Show Reputation Certificate
RP

Fig. 1. Abstract view of Combined Reputation Certificate

The combined certificate offers several advantages; it requires no new infrastructure, is straightforward to implement and requires only one operation to verify the authority's signature. However, it has some drawbacks.

1. The reputation authority is required to issue and manage the certificates besides its routine task to calculate the reputation of the participants.
2. The reputation attribute becomes available to any party who has access to the identity certificate. Users may prefer to hold their reputation values privately except when needed for transactions.
3. Reputation certificates need to be updated frequently so the identity certificate also needs to be issued each time the reputation is updated.

A different way to form a combined certificate was the *smart certificate* proposed by Park and Sandhu [10]. The scheme uses the structure of the X.509 certificate as its basis and the extension fields in the original certificate are used to incorporate additional attributes. Each attribute in the certificate is managed by different authorities. Although the smart certificate has several desirable properties, a major limitation highlighted by Chadwick and Otenko [2] is that it is automatically invalid once any attribute is changed. We expect the reputation rating to change frequently and the certificate needs to be re-issued each time.

3.2 Separate Certificates

The separate certificate corresponds to the autonomic certificate of Park and Sandhu [11]. Figure 2 is a graphical representation of the solution showing the two types of certificates used. The identity certificate is issued by the certificate authority, while the reputation certificate will be issued by the reputation authority. The certificates are linked due to shared information, in particular the unique name (or X.509 *distinguished name*) from the identity certificate may be included in the attribute certificate.

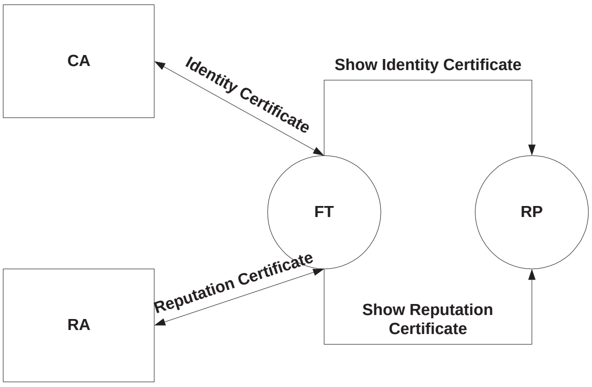


Fig. 2. Abstract view of Separate Certificate

Because there are two authorities, separation of duties can be conducted which can reduce the problem of overloading the reputation authority. The reputation authority is only responsible for the calculation of the reputation, while the certificate authority is responsible for the registration of the feedback targets. The identity certificate is used as an identity mechanism for the feedback target. Like the combined certificate solution, this solution also has its limitations.

1. It is costly to match the identity certificate and the reputation certificate especially to the relying party RP who has to do three steps of verification: first to check the validity of the identity certificate; second to check the validity of the reputation certificate; third to match between these two certificates.
2. RP cannot determine whether RA is authorized to provide reputation for FTs. This means that relying parties have to independently check the policy and practice statements for any issuers of attribute certificates.

3.3 Related Certificates

The idea of related certificates is to ensure that the attribute certificate has a functional link to the identity certificate, beyond simply referring to the same identity. This corresponds to the chained certificates of Park and Sandhu [11]. The difference from the separate certificate option is that now the binding information in the attribute certificate depends on the CA signature on the identity certificate. In other words, the attribute certificate is bound to a specific instance of the identity certificate.

Using an attribute certificate related to the identity certificate as the reputation certificate is a reasonable option. However, the drawbacks already mentioned for separate certificates still apply. Independent signature checking increases the computational burden. The issue of authorization of the RA also applies here, but with a different twist. The issuer of attribute certificate is free to act independently of the $CA$of the identity certificate. However, CAs may object to use of their certificates by third parties for purposes without their consent and may put in place legal obstacles to prevent this.

4 Articulated Certificates

From the discussion in section 3 we see that each of the previous proposals for binding identity and attribute certificates has some drawbacks when used for reputation certificates, although separate certificates or related certificates could be reasonable choices. In this section we proposed a new scheme for linking reputation and identity certificates. We called this an *articulated certificate.*

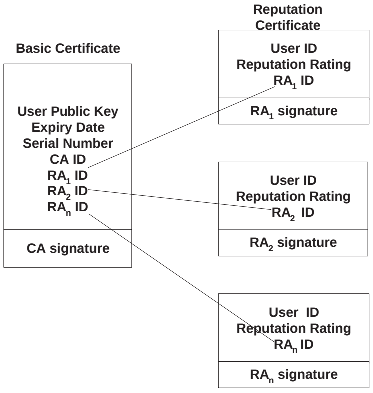


Fig. 3. Abstract View of Articulated Certificate

Figure 3 illustrates the view of the proposed scheme. The properties of articulated certificates are different from all the options considered in section 3.

- Articulated certificates can only be issued by entities that have been authorised to do so by the identity CA. Moreover, the authority to issue may be restricted for a specific purpose or particular time interval.
- The articulated certificate can be verified using the CApublic key alone – no separate certificate is required for the reputation authority.
- The identity certificate may be used either with or without the attribute certificate.

A major feature of our proposed solution is to use the concept of *delegation* to allow the certification authority to give power to the reputation authority to link to the original certificate. Delegation enables RA to update reputation rating in the certificate without invalidating the certificate. The CA delegates his signing capability to RA using the proxy signature scheme. Figure 4 shows the abstract view of the proposed architecture.

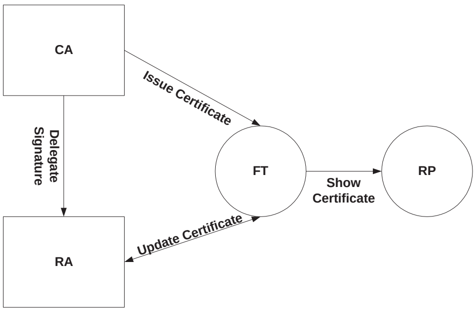


Fig. 4. Proposed Architecture

4.1 Proxy Signatures

Proxy signature schemes allow an original signer to delegate signing capability to another entity, the proxy signer. The first proxy signature scheme was introduced by Mambo et al. [8]. Subsequently a number of schemes have been proposed in the literature [14, 6]. For our purpose the scheme of Mambo et al. (MUO scheme hereafter) will be employed. However, other proxy signature schemes can also be used in our proposal. A brief review of MUO scheme follows.

System Settings: Global system parameters consist of a large prime p, a prime factor q of $p-1$, and an element $g \in Z_p^*$ of order q. Computations take place in Z_p^* unless indicated otherwise. Entity A denotes the original signer and B denotes the proxy signer. Assume that x_A is a private key for A and the corresponding public key $y_A = g^{x_A}$ and m_w is a statement about delegation which typically contains some particulars including the proxy signer identification. A one way hash function $\mathcal{H}()$ is used. The scheme can be divided into four phases; generation of a proxy key, verification of the proxy key, signing using the proxy key and verifying the proxy signature. A small modification is made to the original of MUO scheme to include the hash of m_w.

Generation of a proxy key. A chooses a random number, $k \in_R Z_q^*$ and computes $r = g^k$. He proceeds to compute $s_P = x_A\mathcal{H}(m_w) + kr \bmod q$ and then sends (s_P, r) to B securely.

Verification of the proxy key. Upon receiving (s_P, r), B verifies $g^{s_P} \overset{?}{=} y_A^{\mathcal{H}(m_w)} r^r$. If this equation holds B accepts it is a valid proxy key.

Signing using the proxy key. B signs message m using the proxy key s_P. The signed message is $S(m), r$ where $S()$ is any discrete log signature generation algorithm.

Verifying of the proxy signature. A verifier first calculates $y_P = y_A^{\mathcal{H}(m_w)} r^r$ and checks the validity of the proxy signature $V(y_P, message) \overset{?}{=} true$ where $V()$ is a the signature verification algorithm.

Since MUO scheme is employed, the properties of the scheme of MUO are automatically inherited into our proposal.

1. **Unforgeablity** Besides CA, only RA can create a valid proxy signature. The third parties who are not designated as a proxy signer cannot create a valid proxy key.
2. **Verifiability** RP can be convinced of the original signer agreement on the signed message.
3. **Identifiability** Anyone can determine the identity of the proxy signer from a proxy signature.
4. **Undeniability** Once the proxy signer creates a valid proxy signature he cannot repudiate it.

4.2 Protocol of the Scheme

The protocol consists of six phases as follows. Execution of the phases is not necessarily in sequential order, except that the delegation and registration phases have to be executed prior to the other phases. Some phases may need to be executed more than once, such as updating certificate, showing certificate and validating certificate.

- **Delegation** CA delegates signing capability to RA so that RA can update the certificate of FT with a new reputation rating. It is assumed that both parties CA and RA have already agreed upon the terms and conditions of delegation beforehand which are encoded in m_w. To delegate, CA executes the generation phase of the MUO scheme by choosing a random number k and computes $r_{RA} = g^k$. This is followed by computing $s_{RA} = x_{CA}\mathcal{H}(m_w) + kr_{RA} \bmod q$ and sends (s_{RA}, r_{RA}) to RA securely. On receiving this pair RA verifies $g^{s_{RA}} \overset{?}{=} y_{CA}\mathcal{H}(m_w) r_{RA}^{r_{RA}}$. If this holds RA accepts it is a valid proxy key.
- **Registration** FT creates a public key y_{FT} and the corresponding private key x_{FT}. y_{FT} and ID_{FT} are securely sent to CA for registration. A typical certificate format of the basic certificate may be as follows:

Sig_{CA}	FT	y_{FT}	Exp	CA	RA

  where Sig_{CA} denotes the CA's signature and Exp denotes expiry date of the certificate, On receiving FT's particulars, CA verifies their validity. The certificate is signed by CA using his private key x_{CA} and is sent to FT.

Notice that it is not essential to include the identity of the reputation authorities with the identity certificate, as shown above. Instead, the RA may be identified separately to the relying party by the certificate owner.

- **Sending Identity Certificate** FT is required to send his identity certificate to RA for the initial contact. It can then be recorded in a database maintained by RA until a new identity certificate is issued.
- **Updating Reputation Certificate**
  To prevent unnecessary updating, statistics of activity of the feedback target may be used to determine when the reputation rating should be updated. An active user may be given a short expiry date while an inactive user has a longer one. To issue a new reputation certificate RA signs it using the proxy private key s_{RA}. The certificate is sent to FT. A typical certificate format of the articulated certificate may be as follows.

Sig_{RA}	ExpR	FT	RA	FT Rating

  where Sig_{RA} is the signature of RA and ExpR denotes the expiry date of the reputation rating,
- **Showing Certificate** Before any engagement with the intended RP, FT may be required to show his reputation certificate to RP so that his reputation can be evaluated.
- **Validating Reputation Rating** Prior to accepting the rating in the reputation certificate, RP calculates y_{RA} and verifies the certificate validity based on signature to the conditions in m_w. If so RP accepts the reputation rating as a valid reputation rating.

5 Discussion

There are several advantages held by our scheme compared to other schemes.

- Only one operation is required to verify the reputation certificate, as only the RA's signature needs to be checked by the relying party while the validity of the identity certificate is verified by RA. This advantage is also shared by the basic certificate. Separate certificates and related certificates, on the other hand, require three computations to verify the validity of both certificates.
- There is a separation of duties between the identity CA and reputation authority. This is generally a good security practice, and ensures that neither is overloaded with management tasks.
- Our scheme implements tightly-coupled binding between the identity and reputation certificates because a single identity certificate may be mapped to multiple reputation certificates. This advantage is shared by the combined certificate while the separate certificate implements loosely-coupled solution.
- Our proposal has high reusability because changes to the reputation certificate or the identity certificate cannot invalidate the reputation certificate. This is shared by the separate certificate while the basic reputation and the related certificates have low reusability because any changes invalidate them.

References

1. G. Bolton, E. Katok, and A. Ockenfels. How Effective are Online Reputation Mechanisms? Discussion Papers on Strategic Interaction 25-2002, Max-Planxk-Institut, 2002.
2. D. W. Chadwick and A. Otenko. The permis X.509 role based privilege management infrastructure. *Future Generation Computer Systems*, 19(2):277–289, February 2003.
3. D. Fahrenholtz and W. Lamersdorf. Transactional security for a distributed reputation management system. In *Proceedings of the 3rd International Conference on Electronic Commerce and Web Technologies*, volume 2455 of LNCS, pages 214–223. Springer-Verlag, 2002.
4. M. Gupta, P. Judge, and M. Ammar. A reputation system for peer-to-peer networks. In *ACM 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2003)*, pages 144–152. ACM Press, June 1-3 2003.
5. ITU-T. Recommendation X.509 Information technology - Open systems Interconnection - The Directory : Authentication framework, 1997.
6. B. Lee, H. Kim, and K. Kim. Strong proxy signature and its applications. In *SCIS'2001*, pages 603–608, Jan 23-26 2001.
7. C. Y. Liau, X. Zhou, S. Bressan, and K.-L. Tan. Efficient distributed reputation scheme for peer-to-peer systems. In *The 2nd International Human.Society@Internet Conference*, volume LNCS 2713, pages 54–63. Springer-Verlag, 2003.
8. M. Mambo, K. Usuda, and E. Okamoto. Proxy signatures for delegating signing operation. In *Proceedings of the 3rd ACM conference on Computer and communications security*, pages 48–57. ACM Press, 1996.
9. R. Oppliger, G. Pernul, and C. Strauss. Using attribute certificates to implement role-based authorization and access controls. In *Proceedings of the 4. Fachtagung Sicherheit in Informationssystemen (SIS 2000), Zürich (Switzerland)*, pages 169–184, October 5-6, 2000.
10. J. S. Park and R. Sandhu. Smart certificates: Extending X.509 for secure attribute service on the web. In *Proceedings, 22nd National Information Systems Security Conference*, pages 337–348, October 18-21 1999.
11. J. S. Park and R. Sandhu. Binding identities and attributes using digitally signed certificates. In *16th Annual Computer Security Applications Conference (ACSAC)*, pages 120–127, December 11-15 2000.
12. P. Resnick, R. Zechauser, E. Friedman, and K. Kuwabara. Reputation systems. *Communication of the ACM*, 43(12):45–48, December 2000.
13. P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The value of reputation on eBay: A controlled experiment. In *Working paper presented at the ESA conference*, June 2002.
14. S. P. S. Kim and D. Won. Proxy signatures, revisited. In *Proc. of ICICS'97, International Conference on Information and Communications Security*, volume LNCS 1334, pages 223–232. Springer, 1997.
15. P. R. Zimmermann. *The Official PGP User's Guide.* MIT Press, Cambridge, MA, USA, 1995. Available at http://www-mitpress.mit.edu/mitp/recent-books/comp/pgp-user.html.

A Formal Proof of Security of Zhang and Kim's ID-Based Ring Signature Scheme ⋆

Javier Herranz

Dept. Matemàtica Aplicada IV, Universitat Politècnica de Catalunya
C. Jordi Girona, 1-3, Mòdul C3, Campus Nord, 08034-Barcelona, Spain
jherranz@mat.upc.es
http://www-ma4.upc.es/∼jherranz

Abstract. In this work we provide a formal analysis of the security of an identity-based ring signature scheme proposed by Zhang and Kim in [10]. We first define the security requirements that this kind of schemes must satisfy; or in other words, the capabilities and goals of the most powerful attacks these schemes must remain secure against. Then we prove, in the random oracle model, that the above-mentioned scheme is secure against the defined attacks, assuming that the Computational Diffie-Hellman problem is hard to solve.

1 Introduction

In a *ring signature scheme*, a user computes a signature on behalf of a set (or ring) of users which contains himself. The goal is that any verifier must be convinced that the signature has been computed by some member of this ring, but he has no better way than at random to guess which member is the actual author of the signature.

In practice, if the communications system is authenticated with the use of a Public Key Infrastructure (PKI) based on certificates, the signer must first verify that the public keys of the ring correspond (via a certificate) to the identities of the users that he wants to include on the ring. Later, the verification process of a ring signature obviously employs the public keys of the members of the ring. Therefore, the verifier must first check that these public keys are actually certificated as the ones of the members of the ring.

This means that the cost of both processes of generating and verifying a ring signature substantially increases because of the necessary management of digital certificates. Any possible alternative which avoids the necessity of a PKI is very welcome if we want to design efficient public key cryptosystems, in particular ring signature schemes where the number of certificates that must be checked in each operation can be reasonably high.

Identity-based (from now on, ID-based) cryptography, introduced by Shamir in 1984 [9], is a solution to this problem. The idea is that the public key of a

⋆ This work was partially supported by Spanish *Ministerio de Ciencia y Tecnología* under project TIC 2003-00866.

user can be easily (and publicly) computed from his identity (for example, from a complete name, an e-mail or an IP address). Then, the secret key is derived from the public key. In this way, certificates which link identities and public keys are not needed any more, because anyone can easily verify that some public key PK_U corresponds in fact to user U. The process that generates secret keys from public keys must be executed by an external entity, known as the *master*.

In this work we analyze the security of an ID-based ring signature scheme, based on bilinear pairings. Let us do a brief overview of some works related to ring signatures.

In [8], Rivest, Shamir and Tauman formalize the concept of ring signature schemes, and propose a scheme which they prove existentially unforgeable under adaptive chosen-message attacks, in the ideal cipher model, assuming the hardness of the RSA problem. This scheme also uses a symmetric encryption scheme and the notion of combining functions.

Bresson, Stern and Szydlo show in [3] that the scheme of [8] can be modified in such a way that the new scheme is proved to achieve the same level of security, but under the strictly weaker assumption of the random oracle model.

In [1], Abe, Ohkubo and Suzuki give general constructions of ring signature schemes for a variety of scenarios, including those where signature schemes are based on one-way functions, and those where signature schemes are of the three-move type (for example, Schnorr's signature scheme).

Some security results for generic ring signature schemes, as well as a new specific scheme based on Schnorr's signature scheme, are given by Herranz and Sáez in [5].

Finally, the only ID-based ring signature scheme proposed until now (as far as we know) is the one by Zhang and Kim [10], which is based on pairings. However, they do not provide a formal proof of the existential unforgeability of the proposed scheme.

We provide such a formal proof of security for this ID-based ring signature scheme, assuming that the Computational Diffie-Hellman problem is hard to solve. The proof uses standard techniques in the random oracle model [2], like replaying attacks (formalized in the forking lemmas by Pointcheval and Stern in [7]), which have been already employed to prove the security of other ring signature schemes, for example [1, 5].

2 Zhang and Kim's ID-Based Ring Signature Scheme

In this section we review the ID-based ring signature scheme proposed by Zhang and Kim in [10]. We first explain some basics on bilinear parings and on ring signature schemes.

2.1 A Note on Pairings

Let $\mathbb{G}_1$ be an additive group of prime order q, generated by some element P. Let $\mathbb{G}_2$ be a multiplicative group with the same order q. We consider a *pairing* as a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following three properties:

1. It is bilinear, which means that given elements $T_1, T_2, T_3 \in \mathbb{G}_1$, we have that $e(T_1 + T_2, T_3) = e(T_1, T_3) \cdot e(T_2, T_3)$ and $e(T_1, T_2 + T_3) = e(T_1, T_2) \cdot e(T_1, T_3)$. In particular, for all $a, b \in \mathbb{Z}_q$, we have $e(aP, bP) = e(P, P)^{ab} = e(P, abP) = e(abP, P)$.
2. The map e can be efficiently computed for any possible input pair.
3. The map e is non-degenerate: there exist elements $T_1, T_2 \in \mathbb{G}_1$ such that $e(T_1, T_2) \neq 1_{\mathbb{G}_2}$.

Combining properties 1 and 3, it is easy to see that $e(P, P) \neq 1_{\mathbb{G}_2}$ and that the equality $e(T_1, P) = e(T_2, P)$ implies that $T_1 = T_2$.

The typical way of obtaining such pairings is by deriving them from the Weil or the Tate pairing on an elliptic curve over a finite field. The interested reader is referred to [11] for a complete bibliography of cryptographic works based on pairings.

2.2 Ring Signatures

The idea of a ring signature is the following: a user wants to compute a signature on a message, on behalf of a set (or ring) of users which includes himself. He wants the verifier of the signature to be convinced that the signer of the message is in effect some of the members of this ring. But he wants to remain completely anonymous. That is, nobody will know which member of the ring is the actual author of the signature.

These two informal requirements are ensured, if the scheme satisfies the following properties:

1. **Anonymity:** any verifier should not have probability greater than $1/n$ to guess the identity of the real signer who has computed a ring signature on behalf of a ring of n members. If the verifier is a member of the ring distinct from the actual signer, then his probability to guess the identity of the real signer should not be greater than $1/(n-1)$.
2. **Unforgeability:** among all the proposed definitions of unforgeability (see [4]), we consider the strongest one: any attacker must have negligible probability of success in forging a valid ring signature for some message m on behalf of a ring that does not contain himself, even if he knows valid ring signatures for messages, different from m, that he can adaptively choose.

Ring signatures are a useful tool to provide anonymity in some scenarios. For example, if a member of a group wants to leak to the media a secret information about the group, he can sign this information using a ring scheme. Everybody will be convinced that the information comes from the group itself, but anybody could accuse him of leaking the secret.

A different application is the following: if the signer A of a message wants that the authorship of the signature could be entirely verified only by some specific user B, he can sign the message with respect of the ring $\{A, B\}$. The rest of users could not know who between A and B is the author of the signature, but B will be convinced that the author is A.

2.3 The Scheme

Zhang and Kim proposed in [10] the first ID-based ring signature scheme, following the idea behind the ring signature schemes proposed by Abe, Ohkubo and Suzuki in [1]. We review Zhang and Kim's scheme in this section.

Setup: let $\mathbb{G}_1$ be an additive group of prime order q, generated by some element P. Let $\mathbb{G}_2$ be a multiplicative group with the same order q. We need $q \geq 2^k$, where k is the security parameter of the scheme. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a pairing as defined in Section 2.1. Let $H_1 : \{0,1\}^* \to \mathbb{G}_1 - \{0\}$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_q$ be two hash functions (in the proof of security, we will assume that they behave as random oracles [2]).

The master entity chooses at random his secret key $x \in \mathbb{Z}_q^*$ and publishes the value $Y = xP \in \mathbb{G}_1$.

Secret key extraction: a user U, with identity $ID_U \in \{0,1\}^*$, has public key $PK_U = H_1(ID_U)$. When he requests the master for his matching secret key, he obtains the value $SK_U = xPK_U$.

Ring signature: consider a ring $\mathcal{U} = \{U_1, \ldots, U_n\}$ of users; for simplicity we denote $PK_i = PK_{U_i} = H_1(ID_{U_i})$. If some of these users U_s, where $s \in \{1, \ldots, n\}$, wants to anonymously sign a message m on behalf of the ring $\mathcal{U}$, he acts as follows:

1. Choose a random $T \in \mathbb{G}_1$ and compute $c_{s+1} = H_2(\mathcal{U}, m, e(T, P))$.
2. For $i = s+1, \ldots, s-1$ (where i is considered modulo n), choose T_i at random in $\mathbb{G}_1$. Compute $c_{i+1} = H_2(\mathcal{U}, m, e(T_i, P) \cdot e(c_i PK_i, Y))$.
3. Compute $T_s = T - c_s SK_s \bmod q$.
4. Define the signature of the message m made by the ring $\mathcal{U} = \{U_1, \ldots, U_n\}$ to be $(\mathcal{U}, m, c_0, T_0, T_1, \ldots, T_{n-1})$.

Verification: the validity of the signature is verified by the recipient of the message in the following way:

1. For $i = 0, 1, \ldots, n-1$, compute $c_{i+1} = H_2(\mathcal{U}, m, e(T_i, P) \cdot e(c_i PK_i, Y))$.
2. Accept the signature as valid if $c_n = c_0$, and reject it otherwise.

By using the bilinear property of the pairing e, it is easy to see that the scheme is correct.

3 A Formal Security Analysis

In their paper [10], Zhang and Kim do not provide a formal proof of the security of this scheme. Their arguments are quite heuristic or intuitive. They can be enough for anonymity, but not for unforgeability. For example, they do not define the capabilities of an adversary against an ID-based ring signature scheme. They assert that the scheme is secure because in the case $n = 1$ the scheme is exactly the ID-based signature scheme proposed by Hess in [6], and since this scheme is proved to be secure, then the ring signature scheme is also secure. Clearly, this argument is not enough. We give in this section a formal proof of the security of their scheme, which employs some standard techniques, like replaying attacks [7], already used to prove the security of other ring signature schemes [1, 5].

3.1 The Security Model

We must consider the most powerful attack against an ID-based ring signature scheme, that we call *chosen message and identities attack*. Such an attacker $\mathcal{A}$ is allowed to:

- make Q_1 queries to the random oracle H_1 and Q_2 queries to the random oracle H_2;
- ask for the secret key of Q_e identities of its choice (extracting oracle);
- ask Q_s times for valid ring signatures, on behalf of rings of its choice, of messages of its choice (signing oracle).

The total number of queries must be polynomial in the security parameter. The attacker is successful if it outputs, in polynomial time and with non-negligible probability, a valid ring signature for some message m and some ring of users $\mathcal{U} = \{U_1, \ldots, U_n\}$ such that:

- the attacker has not asked for the secret key of any of the members of the ring $\mathcal{U}$;
- the attacker has not asked for a valid ring signature, on behalf of the ring $\mathcal{U}$, of message m.

3.2 The Computational Diffie-Hellman Problem

We consider the following well-known problem in the group $\mathbb{G}_1$ of prime order q, generated by P.

Definition 1. *Given the elements $P, aP, bP \in \mathbb{G}_1$, for some random values $a, b \in \mathbb{Z}_q^*$, the Computational Diffie-Hellman (CDH) problem consists of computing the element abP.*

The Computational Diffie-Hellman Assumption asserts that, if the order of $\mathbb{G}_1$ is $q \geq 2^k$, then any polynomial time algorithm that solves the CDH problem has a success probability p_k which is negligible in the security parameter k. In other words, for all polynomial $f()$, there exists an integer k_0 such that $p_k < \frac{1}{f(k)}$, for all $k \geq k_0$.

3.3 Proving the Unforgeability of the Scheme

We start with a technical lemma which will be necessary for the proof of the main result. Its proof can be found in [7].

Lemma 1. *(The Splitting Lemma) Let* $A \subset X \times Y$ *such that* $\Pr[(x,y) \in A] \geq \delta$. *For any* $\alpha < \delta$, *define*

$$B = \{(x,y) \in X \times Y | \Pr_{y' \in Y}[(x,y') \in A] \geq \delta - \alpha\}.$$

Then the following statements hold:

1. $\Pr[B] \geq \alpha$.
2. *For any* $(x,y) \in B$, $\Pr_{y' \in Y}[(x,y') \in A] \geq \delta - \alpha$.
3. $\Pr[B|A] \geq \alpha/\delta$.

We prove now that the existence of a successful attack against the ID-based ring signature scheme could be used to solve the Computational Diffie-Hellman problem in $\mathbb{G}_1$ (a proof by reduction). Since this problem is assumed to be hard, we conclude that there does not exist such an attack. In this way, the scheme is proved to be existentially unforgeable under chosen message and identities attacks.

In this proof, we assume that the hash functions H_1 and H_2 behave as random oracles [2].

Theorem 1. *Let* k *be a security parameter, and let the order of* $\mathbb{G}_1$ *be* $q \geq 2^k$. *Let* $\mathcal{A}$ *be a probabilistic polynomial time Turing machine attacking the considered ID-based ring signature scheme. We denote by* Q_1, Q_2, Q_e *and* Q_s *the number of queries that* $\mathcal{A}$ *can ask to the random oracles* H_1 *and* H_2 *and to the extracting and signing oracles, respectively. We denote by* N *the maximum cardinality of the rings for which* $\mathcal{A}$ *asks for a valid signature.*

Assume that $\mathcal{A}$ *produces, within polynomial time* t *and with non-negligible probability of success* ε, *a valid ring signature* $(\mathcal{U}, m, c_0, T_0, T_1, \ldots, T_{n-1})$, *such that* $\mathcal{A}$ *has not asked for the secret key of any of the members of* $\mathcal{U}$, *and has not asked for a valid ring signature of* m *on behalf of the ring* $\mathcal{U}$. *Assume that* $q > \max\{(Q_1 + Q_e)^2, 2N, 2Q_2Q_s\}$ *and that* $\varepsilon > \frac{64\ Q_2^2}{q}$.

Then the Computational Diffie-Hellman problem in $\mathbb{G}_1$ *can be solved with probability* $\varepsilon' \geq \frac{9}{100\ Q_1}$ *and in time* $t' \leq \frac{64 Q_2^2}{\varepsilon} t$.

Proof. Let (P, aP, bP) be an input of the CDH problem in $\mathbb{G}_1$, for some random $a, b \in \mathbb{Z}_q^*$. We design a solver algorithm $\mathcal{B}$ that uses the attacker $\mathcal{A}$ as a subroutine, and finds the solution of the CDH problem.

First, $\mathcal{B}$ runs the setup phase of the ID-based ring signature scheme, defining the public master key as $Y = aP$. Then $\mathcal{B}$ runs the attacker $\mathcal{A}$. The algorithm $\mathcal{B}$ must simulate the environment of the attacker $\mathcal{A}$; that is, it must provide consistent answers to all the queries that $\mathcal{A}$ is allowed to make (random oracles H_1 and H_2, extracting and signing oracles).

Furthermore, $\mathcal{B}$ chooses at random a value $\ell \in \{1, 2, \dots, Q_1\}$. When the attacker $\mathcal{A}$ makes the ℓ-th query to the random oracle H_1, with some identity ID_ℓ, the algorithm $\mathcal{B}$ sets $PK_\ell = H_1(ID_\ell) = bP$, and sends this value to the attacker. Later, if the attacker $\mathcal{A}$ asks for the secret key of ID_ℓ to the extracting oracle, then the algorithm $\mathcal{B}$ stops and outputs "fail".

For the rest of identities $\{ID_j\}_{1 \le j \le Q_e + Q_1}$ that $\mathcal{A}$ queries to the extracting oracle or to the random oracle H_1, $\mathcal{B}$ can provide consistent answers as follows: $\mathcal{B}$ chooses a random element $x_j \in \mathbb{Z}_q^*$ and computes the values $PK_j = x_j P$ and $SK_j = x_j Y$, where Y is the master public key. Then $\mathcal{B}$ sets $H_1(ID_j) = PK_j$, and stores this relation in a random oracle list for H_1. If the query was a random oracle query, $\mathcal{B}$ sends to $\mathcal{A}$ the value PK_j. If the query was an extracting query, $\mathcal{B}$ sends to $\mathcal{A}$ the value SK_j for the secret key, as well.

The only inconsistency problem happens if two different executions (with different identities ID_i and ID_j) of this simulation result in the same value $PK_i = PK_j$. The probability of such a collision is, however, less than $\frac{(Q_1+Q_e)^2}{2} \cdot \frac{1}{q}$.

On the other hand, every time that $\mathcal{A}$ asks for a valid ring signature for a message m and a ring $\mathcal{U}$, the algorithm $\mathcal{B}$ proceeds as follows:

1. Choose at random $c_0 \in \mathbb{Z}_q$.
2. For $i = 0, 1, \dots, n-1$, choose T_i at random in $\mathbb{G}_1$. If $i \neq n-1$, compute $c_{i+1} = H_2(\mathcal{U}, m, e(T_i, P) \cdot e(c_i PK_i, Y))$. In order to compute this value, the algorithm $\mathcal{B}$ constructs, as before, a random oracle list for H_2. If the input is already in the list, it outputs the matching value. If not, it chooses a random value in $\mathbb{Z}_q$, outputs it and stores the new relation in the list.
3. Define $H_2(\mathcal{U}, m, e(T_{n-1}, P) \cdot e(c_{n-1} PK_{n-1}, Y))$ to be c_0. Store this relation in the list for H_2.
4. Send the tuple $(\mathcal{U}, m, c_0, T_0, T_1, \dots, T_{n-1})$ to $\mathcal{A}$.

For the queries of $\mathcal{A}$ to the random oracle H_2, the algorithm $\mathcal{B}$ proceeds in the same way: it looks for the input in the list, outputting the matching value if it finds it, or a random value otherwise. Now the risk is that, in step 3 of the above simulation process, the obtained tuple $(\mathcal{U}, m, e(T_{n-1}, P) \cdot e(c_{n-1} PK_{n-1}, Y))$ has been already queried by $\mathcal{A}$ to the random oracle H_2. The probability of such a collision is less than $\frac{Q_2}{q}$ for each execution of the signature simulation, and so less than $\frac{Q_s Q_2}{q}$ for the whole process.

Summing up, the algorithm $\mathcal{B}$ successfully simulates the environment of $\mathcal{A}$ with probability greater than $\epsilon_1 = (1 - \frac{(Q_1+Q_e)^2}{2q})(1 - \frac{Q_s Q_2}{q})$.

We denote by ω the whole set of random tapes that take part in an attack by $\mathcal{A}$, with the environment simulated by $\mathcal{B}$, but excluding the randomness related to the oracle H_2. The success probability of $\mathcal{A}$ in forging a valid ring signature scheme is then taken over the space (ω, H_2). If we denote by $\mathcal{S}$ the set of successful executions of $\mathcal{A}$, we have that $\Pr[(\omega, H_2) \in \mathcal{S}] \ge \varepsilon$.

Now consider a ring signature $(\mathcal{U}, m, c_0, T_0, T_1, \dots, T_{n-1})$ forged by $\mathcal{A}$. We denote as R_i the value $e(T_i, P) \cdot e(c_i PK_i, Y)$, for all $i = 0, \dots, n-1$. We use the notation $\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_{Q_2}$ for the different queries that $\mathcal{A}$ makes to the random oracle H_2. By the ideal randomness of this oracle, the probability that $\mathcal{A}$ has not asked for some of the tuples $(\mathcal{U}, m, R_i)$, with $i = 0, \ldots, n-1$ (and so $\mathcal{A}$ must have guessed the corresponding output), is less than $\frac{n}{q} \leq \frac{N}{q}$.

We refer as $\mathcal{S}'$ to the successful executions of $\mathcal{A}$, with $\mathcal{B}$ simulating its environment, where $\mathcal{A}$ has queried all the tuples $(\mathcal{U}, m, R_i)$ in the forged signature to the random oracle H_2. We have that $\epsilon_2 = \Pr[(\omega, H_2) \in \mathcal{S}'] \geq \varepsilon\ \epsilon_1(1 - \frac{N}{q})$. The restriction on the values of q, Q_1, Q_2, Q_e and Q_s in the statement of this theorem implies that $\epsilon_2 > \varepsilon/8$.

Because of the ring structure formed by the queries that $\mathcal{A}$ makes to the random oracle H_2, there exists at least one index $k \in \{1, 2, \ldots, n\}$ such that the query $\mathcal{Q}_u = (\mathcal{U}, m, R_k)$ was made to H_2 before the query $\mathcal{Q}_v = (\mathcal{U}, m, R_{k-1})$ (that is, $u < v$). This pair (u, v) is called then a gap index. If there are two or more gap indexes in a forged signature, we consider only the one with the smallest value u. This allows us to define the subset $\mathcal{S}'_{u,v}$ of $\mathcal{S}'$ as the set of executions in $\mathcal{S}'$ whose gap index is (u, v). This gives us a partition of $\mathcal{S}'$ in exactly $\frac{Q_2(Q_2+1)}{2}$ classes.

If $\mathcal{B}$ invokes $t_1 = 1/\epsilon_2$ times the attacker $\mathcal{A}$ with randomly chosen (ω, H_2), it obtains a successful execution $(\tilde{\omega}, \tilde{H}_2) \in \mathcal{S}'_{u,v}$, for some gap index (u, v), with probability $1 - (1 - \epsilon_2)^{1/\epsilon_2} = 1 - \left[\left(1 + \frac{1}{-1/\epsilon_2}\right)^{-1/\epsilon_2}\right]^{-1} \geq 1 - e^{-1} > 3/5$.

Now we define the set of gap indexes which are more likely to appear as

$$I = \{(u, v) \text{ s.t. } \Pr[(\omega, H_2) \in \mathcal{S}'_{u,v} \mid (\omega, H_2) \in \mathcal{S}'] \geq \frac{1}{Q_2(Q_2+1)}\}.$$

And the corresponding subset of successful executions as $\mathcal{S}'_I = \{(\omega, H_2) \in \mathcal{S}'_{u,v}$ s.t. $(u, v) \in I\}$.

It holds that $\Pr[(\omega, H_2) \in \mathcal{S}'_I \mid (\omega, H_2) \in \mathcal{S}'] \geq 1/2$. In effect, since the sets $\mathcal{S}'_{u,v}$ are disjoint, we have

$$\Pr[(\omega, H_2) \in \mathcal{S}'_I \mid (\omega, H_2) \in \mathcal{S}'] = \sum_{(u,v) \in I} \Pr[(\omega, H_2) \in \mathcal{S}'_{u,v} \mid (\omega, H_2) \in \mathcal{S}'] =$$

$$1 - \sum_{(u,v) \notin I} \Pr[(\omega, H_2) \in \mathcal{S}'_{u,v} \mid (\omega, H_2) \in \mathcal{S}'].$$

Since the complement of I contains at most $\frac{Q_2(Q_2+1)}{2}$ gap indexes, we have that this probability is greater than $1 - \frac{Q_2(Q_2+1)}{2} \cdot \frac{1}{Q_2(Q_2+1)} = 1/2$. Therefore, with probability at least $1/2$, the specific successful execution $(\tilde{\omega}, \tilde{H}_2)$ is in $\mathcal{S}'_I$.

Consider any possible likely gap index $(u, v) \in I$; we have that

$$\Pr[(\omega, H_2) \in \mathcal{S}'_{u,v}] = \Pr[(\omega, H_2) \in \mathcal{S}'] \cdot \Pr[(\omega, H_2) \in \mathcal{S}'_{u,v} \mid (\omega, H_2) \in \mathcal{S}'] \geq$$

$$\geq \epsilon_2 \cdot \frac{1}{Q_2(Q_2+1)}.$$

We split H_2 as (H'_2, c_k), where H'_2 corresponds to the answers of all the queries to H_2 except the query $\mathcal{Q}_v$, whose answer is denoted as c_k. We apply the Splitting

Lemma (lemma 1), taking $X = (\omega, H'_2)$, $Y = c_k$, $A = \mathcal{S}'_{u,v}$, $\delta = \frac{\epsilon_2}{Q_2(Q_2+1)}$ and $\alpha = \frac{\epsilon_2}{2Q_2(Q_2+1)}$. The lemma says that there exists a subset of executions $\Omega_{u,v}$ such that:

$$\Pr[(\omega, H_2) \in \Omega_{u,v} \mid (\omega, H_2) \in \mathcal{S}'_{u,v}] \geq \frac{\alpha}{\delta} = \frac{1}{2}$$

and such that, for any $(\omega, H_2) \in \Omega_{u,v}$:

$$\Pr_{c'_k}[(\omega, H'_2, c'_k) \in \mathcal{S}'_{u,v}] \geq \delta - \alpha = \frac{\epsilon_2}{2Q_2(Q_2+1)}.$$

Assuming that the concrete execution $(\tilde{\omega}, \tilde{H'}_2, \tilde{c}_k)$ is in $\mathcal{S}'_I$, for some concrete gap index $(\tilde{u}, \tilde{v}) \in I$, then with probability greater than 1/2, the execution is also in $\Omega_{\tilde{u},\tilde{v}}$. In this case, if we now repeat $t_2 = \left(\frac{\epsilon_2}{2Q_2(Q_2+1)} - \frac{1}{q}\right)^{-1}$ times the attack $\mathcal{A}$ with fixed $(\tilde{\omega}, \tilde{H'}_2)$ and randomly chosen $c'_k \in \mathbb{Z}_q$, we obtain with probability again greater than 3/5 a new c'_k such that $(\tilde{\omega}, \tilde{H'}_2, c'_k) \in \mathcal{S}'_{\tilde{u},\tilde{v}}$ and such that $c'_k \neq \tilde{c}_k$.

Since we have imposed in the stating of the theorem that $\varepsilon > \frac{64Q_2^2}{q}$, we have in particular that $\frac{\epsilon_2}{2Q_2(Q_2+1)} > \frac{2}{q}$, which implies that $t_2 < \frac{4Q_2(Q_2+1)}{\epsilon_2}$.

The total probability is then $\epsilon_3 \geq \frac{3}{5} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{3}{5} = \frac{9}{100}$, and the polynomial number of repetitions of the attack $\mathcal{A}$ is

$$t_1 + t_2 < \frac{1}{\epsilon_2} + \frac{4Q_2(Q_2+1)}{\epsilon_2} < \frac{8}{\varepsilon} + \frac{8 \cdot 4 \cdot Q_2 \cdot 2Q_2}{\varepsilon} = \frac{64Q_2^2 + 8}{\varepsilon}.$$

Now consider the two successful executions of the attack $(\tilde{\omega}, \tilde{H'}_2, \tilde{c}_k)$ and $(\tilde{\omega}, \tilde{H'}_2, c'_k)$ that the algorithm $\mathcal{B}$ has obtained. Since the random tapes and H_1 are identical, and the answers of the random oracle H_2 are the same until the query $\mathcal{Q}_{\tilde{v}} = (\mathcal{U}, m, R_{k-1})$, we have in particular that the query $\mathcal{Q}_{\tilde{u}} = (\mathcal{U}, m, R_k)$, which happens before $\mathcal{Q}_{\tilde{v}}$, is also identical for the two executions. Therefore,

$$R_k = e(T_k, P) \cdot e(\tilde{c}_k PK_k, Y) = e(T'_k, P) \cdot e(c'_k PK_k, Y), \text{ with } c'_k \neq \tilde{c}_k.$$

On the other hand, with probability $1/Q_1$, the choice of the index ℓ made by $\mathcal{B}$ is a correct guess, and the public key PK_ℓ corresponds precisely to this PK_k. In particular, this means that the attacker $\mathcal{A}$ has not asked for the secret key matching with PK_ℓ, and so the CDH-solver $\mathcal{B}$ has not output “fail”.

Summing up, with probability $\varepsilon' \geq \frac{9}{100\ Q_1}$ and in time $t' \leq \frac{64Q_2^2+8}{\varepsilon} t$, the algorithm $\mathcal{B}$ obtains values $T_k, T'_k, \tilde{c}_k, c'_k$ such that $e(T_k, P) \cdot e(\tilde{c}_k PK_k, Y) = e(T'_k, P) \cdot e(c'_k PK_k, Y)$, where $PK_k = PK_\ell = bP$ and $Y = aP$.

Since the pairing e is bilinear and non-degenerate, the previous equality implies that $e(T_k + \tilde{c}_k abP, P) = e(T'_k + c'_k abP, P)$ and so $T_k - T'_k = (c'_k - \tilde{c}_k)abP$. Since $c'_k \neq \tilde{c}_k$, one can compute the inverse of $c'_k - \tilde{c}_k$ modulo q, and therefore $\mathcal{B}$ obtains the solution of the CDH problem:

$$abP = \frac{1}{c'_k - \tilde{c}_k}\ (T_k - T'_k)\ \in \mathbb{G}_1.$$

□

Assuming that the Computational Diffie-Hellman problem cannot be solved in polynomial time and with non-negligible probability, this theorem implies that the Zhang and Kim's ID-based ring signature scheme is unforgeable under chosen message and identities attack.

4 Conclusions

In this work we provide a formal model to analyze the unforgeability of ID-based ring signature schemes, by defining the goals and the capabilities of an adversary against such a scheme. Then we prove that the scheme proposed by Zhang and Kim in [10] achieves this level of security, in the random oracle model.

In some way this result completes the work of Zhang and Kim. They designed the scheme and showed that it is unconditionally anonymous, but did not formally prove its unforgeability.

Furthermore, the new formal security model could be used to analyze the security of future proposals of ID-based ring signature schemes.

References

1. M. Abe, M. Ohkubo and K. Suzuki. $1-$out$-$of$-n$ signatures from a variety of keys. *Advances in Cryptology-Asiacrypt'02*, LNCS **2501**, Springer-Verlag, pp. 415–432 (2002).
2. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. *First ACM Conference on Computer and Communications Security*, pp. 62–73 (1993).
3. E. Bresson, J. Stern and M. Szydlo. Threshold Ring Signatures for Ad-hoc Groups. *Advances in Cryptology-Crypto'02*, LNCS **2442**, Springer-Verlag, pp. 465–480 (2002).
4. S. Goldwasser, S. Micali and R. Rivest. A digital signature scheme secure against adaptative chosen-message attacks. *SIAM Journal of Computing*, **17 (2)**, pp. 281–308 (1988).
5. J. Herranz and G. Sáez. Forking lemmas for ring signature schemes. *Proceedings of Indocrypt'03*, LNCS **2904**, Springer-Verlag, pp. 266–279 (2003).
6. F. Hess. Efficient identity based signature schemes based on pairings. *Proceedings of SAC'02*, LNCS **2595**, Springer-Verlag, pp. 310–324 (2002).
7. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, Vol. **13** (3), pp. 361–396 (2000).
8. R. Rivest, A. Shamir and Y. Tauman. How to leak a secret. *Advances in Cryptology-Asiacrypt'01*, LNCS **2248**, Springer-Verlag, pp. 552–565 (2001).
9. A. Shamir. Identity-based cryptosystems and signature schemes. *Advances in Cryptology-Crypto'84*, LNCS **196**, pp. 47–53 (1984).
10. F. Zhang and K. Kim. ID-base blind signature and ring signature from pairings. *Advances in Cryptology-Asiacrypt'02*, LNCS **2501**, Springer-Verlag, pp. 533–547 (2002).
11. The Pairing-Based Crypto Lounge. Web page maintained by Paulo Barreto: `http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html`

B2C Internet Authentication Method using Statistical Keystroke Biometrics

Marino Tapiador[1], Juan A. Sigüenza[2]

[1] IBM Global Services and Universidad Autónoma de Madrid,
Escuela Politécnica Superior, Spain
marino_tapiador@es.ibm.com
[2] Universidad Autónoma de Madrid,
Escuela Politécnica Superior, Spain
jalberto.siguenza@ii.uam.es

Abstract. This paper is focused on biometric user authentication on the Internet using keystroke dynamics. This work describes the advantages of using keystroke biometrics in B2C Internet applications, and also it shows the problems and techniques related to sampling the typing pattern in the Internet. This research work is presented applied to a real experimental system that implements a keystroke dynamics login mechanism based on interkey and holdkey typing times. Several experiments are described and as a result of these experiments, a statistical pattern-recognition model based on mixed typing times and the average normalization technique is presented in this work. Internet open platform technologies are used.

1 Introduction

In the work presented in this paper, we have developed a system based on behavioral biometrics (see [1]) to provide authentication in a secure Internet application: a business-to-customer (B2C) e-commerce system. The system uses the keystroke dynamics to learn which is the user's behavior when he/she types a sequence of fixed characters in the keyboard. Several works on keystroke dynamics authentication can be found, but not considering the specific characteristics of typing in Internet applications: for instance, the preliminary work of Gaines and Lisowski [2], the password experiments of Monrose and Rubin [3], or the authentication studies of Obaidat and Sadoun [4].

Keyboard is a common hardware that comes in useful to authenticate a user in the Internet because nowadays is the common input device in each terminal in the network. This is a key point for e-commerce systems which main objective is to get a big number of customers i.e. in B2C systems. It also implies no extra-hardware needed, that also increases the number of potential users of the system in the Internet. This kind of biometric control in a web application already has a minimal cost because it only needs new software, not extra-hardware.

2 Method

2.1 Data Collection Mechanism for Typing Samples

A ‘sample’ in keystroke biometrics can be information related to: typing difficulty, interkey times (latencies), holdkey times (durations) or others like the number of keys involved on the character generation, keystroke overlaps, etc. Our work is focused on measuring typing times i.e. latencies and durations. In the Internet the problem is to do this sampling process in a platform-independent manner i.e. independently of the CPU frequency where the user is typing.

The prototype developed in this work is based on machine cycles in order to be able to fetch the holdkey times and get maximum sampling precision. Machine cycles can be measured using low-level programming languages or assembler. However this kind of languages depend on the hardware platform –kind of computer- been used, so that it is not a valid approach for a B2C Internet application. This type of application is going to run in the Internet where a lot of different hardware platforms are connected. This requirement implies to use an open-platform language to capture the keystroke biometric data: Java language. With this high-level language we can measure the machine cycles in an indirect way, using machine pseudocycles using an event-based technique where a counter-thread is continuously increasing the value of a variable i.e. this value is the number of machine cycles multiplied by a constant factor equal to the number of machine instructions involved in the adding loop implemented in Java language.

The machine pseudocycles are also dependent on the computer frequency where the user is typing. The idea is we also need to identify the user independently which computer is been used by him/her. In the Internet the user could be using our application from different terminals at different times. Therefore, in order to assure a pattern-recognition algorithm is able to identify the genuine user between different hardware platforms, the biometric samples need to be normalized to be CPU frequency-independent. In this case, a normalization technique by the average value is used. It means the average time of the typed sequence is used as reference time to normalize all the times in the sequence.

2.2 Statistical Model for Typing Recognition

The prototype is a typical three-tier Internet architecture. In our experiments the keystroke pattern-recognition prototype consist of a database with biometric information that is checked with a server program (CGI in C++) when a user does a login into a simulated web system i.e. an HTML page with a Java applet.

When the user tries to access to the website, the system sends a form (the Java applet) where the user must type his/her userId and password. At the same time the system is registering this information, it is also registering biometric data related to the keystroke dynamics of the user, and this information is sent to the server through Internet. The server receives the biometric data and runs a CGI program with the logic related to the pattern-recognition algorithms to try to identify or authenticate to the user by his/her typing rhythm. The CGI compares the information with the information stored into a database of files in the hard disk. This files come from the user registration process in the system (training).

A statistical model provides the pattern-recognition mechanism, and it consists of a parameter-based estimation. We can consider each holdkey or interkey time as the result of a random experiment. That result is a normalized -by the average- time value t in $\Omega = (0, +\infty)$. Let be the hypothesis "each user types with some regularity rate", we can expect the time values generated by the user will be concentrated around some average value, and with small deviations from it. These deviations are minimum when the user is very regular. That behavior is well-modeled by a gaussian function i.e. a normal density $N(\mu,\sigma^2)$ with μ average and σ^2 variance. The analytic expression of this function is well known:

$$N(\mu,\sigma^2) \rightarrow f(x) = (1/\sqrt{2}\ \sigma^2\pi)\ \exp(\ (-1/2\ \sigma^2) \cdot (x- \mu)^2\ ) \qquad \mathbf{(1)}$$

The f(x) corresponds to a normal distribution but we don't know its exact parameters (μ,σ^2). The technique of maximum likelihood shows that the best estimators are:

$$\mu = (1/n)\ \Sigma\ x$$
$$\sigma^2 = (1/n)\ \Sigma\ (x - \mu\ )^2 \qquad \mathbf{(2)}$$

Thus with this technique we get a punctual density estimation following the time distribution for a specific typing. Let be a T_i time which estimated distribution is $N_i(\mu,\sigma^2)$, the probability of getting that T_i is defined by the density $f_i(T_i\ ;\mu,\sigma^2)$. This T_i point probability is used as a scoring function for the T_i point by the system.

During the training phase (ten samples) the user generates the samples composing a template that consist of the sample estimators for the average μ and variance σ^2 for each time interval in the characters sequence. In production time, when a new T_i is generated in a sequence, its probability to occur is calculated using a maximum likelihood estimation by the sample estimators stored into the cited template. Therefore that probability P_i for T_i is obtained based on its distribution estimation $N_i(\mu,\sigma^2)$ calculated during the training phase i.e. $f_i(T_i\ ;\mu,\sigma^2) = P_i$. If P_i is too low it means is highly probability the user has not generated the time, so he/she is a potential fake. The opposite implies the user is probably the genuine user. In order to get this evaluation in a [0,1] rate, it is normalized the density function by its maximum value and it generates this scoring function:

$$S(x) = (1\ /\ S_{max})\cdot\ f_i(x;\ \mu,\sigma^2)\ ,$$
$$S(x) = (\sqrt{2}\ \sigma^2\pi)\cdot\ (1/\sqrt{2}\ \sigma^2\pi)\ \exp(\ (-1/2\ \sigma^2) \cdot (x- \mu)^2\ )\ , \qquad \mathbf{(3)}$$
$$S(x) = \exp(\ (-1/2\ \sigma^2) \cdot (x- \mu)^2\ )$$

During the training phase the sample estimators included in the statistical template are calculated each time using this procedure:

(1st) Average, sample estimator:

$$\mu_1 = (1 / k) \cdot (n \cdot \mu_0 + X_k),\ k = n + 1;\ \ \mu_0 = (1 / n)\ \Sigma\ X_i,\ \ i = 1,2, ...\ n \qquad \textbf{(4)}$$

(2nd) Variance, sample estimator:

$$\sigma_1^2 = (\ \ (1 / (n + 1))\ \Sigma\ X_i^2\ ) - \mu_1^2,\ i = 1,2, ...\ n + 1;$$
$$\sigma_0^2 = (1 / n)\ \Sigma\ (\ X_i - \mu_0)^2\ = (\ (1 / n)\ \Sigma\ X_i^2\ ) - \mu_0^2,\ i = 1,2, ...\ n \qquad \textbf{(5)}$$

Where μ_1 is known because it was previously calculated, and the other term can be also calculated due it depends on $\Sigma\ X_i^2$ and occurs that:

$$\sigma_0^2 = (1 / n)\ \Sigma\ X_i^2 - \mu_0^2 \rightarrow (\sigma_0^2 + \mu_0^2) \cdot n = \Sigma\ X_i^2,\ i = 1,2, ...\ n;\ \text{and thus:}$$
$$\sigma_1^2 = (\ ((\sigma_0^2 + \mu_0^2) \cdot n + X_k^2) / k\ \ ) - \mu_1^2,\ k = n + 1 \qquad \textbf{(6)}$$

So this equations are used to calculate again the sample estimators for the average and the variance each time a new sample is received during the training process, and starting from an initial sample corresponding to the user registration into the system. That first sample is used for the initial averages, and the initial variances are set to zero. This statistical model is used in the same way for interkey times and for holdkey times, just considering the different sequence lengths.

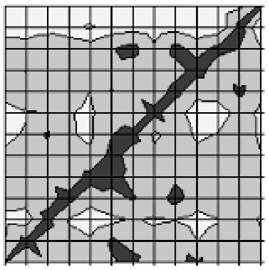
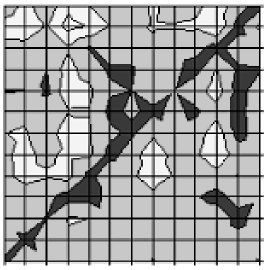
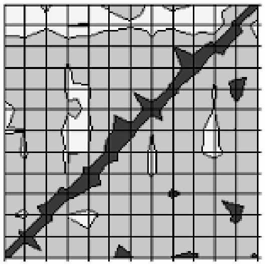
3 Experimental results

In the experiment the samples are composed of interkey times only, or holdkey times only, or both of them i.e. mixed times. The statistical recognition model previously described was used with one simple userId/password sequence (“autonoma” / “internet”). Previous work in this area showed a better recognition rate using simple character sequences (see [5]). The main objective was to compare interkey times versus holdkey times, and mixed times, and to measure the recognition rates. The experiment was performed with a group of men and women with ages between 21 and 48 years old. Netscape and MS-IExplorer v4+ were used under Windows platform.

The Table 1 resumes the results of the experiment and allows comparing the performance of using the three time approaches. The table shows that comparatively the recognition performance is better for mixed times. The figures in Table 1 show the real user typing versus all the possible users included him/her, therefore each figure should present a ‘diagonal effect’ i.e. a diagonal of maximum score values: diagonal points are where the real user is the supposed user. The best performance is associated to the mixed times figure in the sense of it has a more clearly defined ‘diagonal effect’.

Table 1. Experiment results of the three kind of times.

■ 0,9-1,2 ▨ 0,6-0,9 □ 0,3-0,6 □ 0-0,3 = Scoring. (X-axis: real user. Y-axis: supposed user.)

Statistical Model		
Interkey times	Holdkey times	Mixed times
		

Thus, considering the winner mixed times only, the Table 2 illustrates with more detail the scoring rates (%) achieved by this recognition model. The real users typing are represented by the ‘X’ items and the supposed users or templates used are the ‘Y’ items. The ‘diagonal effect’ can be clearly observed and only one error is presented in cell Y12-X9 i.e. a case where the maximum recognition value is obtained with a user different to the genuine user.

Table 2. Experiment results of mixed times.

%	***Y1***	***Y2***	***Y3***	***Y4***	***Y5***	***Y6***	***Y7***	***Y8***	***Y9***	***Y10***	***Y11***	***Y12***	***Y13***
X1	62	38	53	46	47	55	56	37	56	58	47	54	56
X2	41	70	52	46	44	62	57	47	54	49	49	57	48
X3	35	30	55	33	28	37	39	34	40	33	36	53	37
X4	52	48	53	65	56	54	58	46	61	53	56	54	52
X5	48	44	49	41	71	57	53	40	52	48	53	46	47
X6	52	47	59	45	47	80	61	44	61	52	41	58	53
X7	42	42	65	34	39	62	71	42	57	48	36	65	47
X8	36	43	44	42	39	57	48	66	51	46	39	56	42
X9	46	42	62	38	39	62	61	49	66	51	39	69	50
X10	53	47	62	40	51	61	59	45	57	70	46	54	56
X11	54	52	54	51	59	56	61	46	62	56	69	57	46
X12	30	27	42	22	21	39	43	37	39	33	25	76	32
X13	36	31	39	28	34	39	43	35	39	37	23	40	62

4 Conclusions

Considering the final results resumed in Table 1, the accuracy obtained is 99% for mixed times versus a 97% for interkey times and a 94% for holdkey times. Thus, it is observed that mixed times work better than each other separately. In terms of False Accept Rate (FAR) and False Rejection Rate (FRR) the system showed a 0.6% of FAR and a 7% of FRR (for details on FRR/FAR see the work of Wayman [6]).

The experiments suggest that keystroke biometric devices can be developed for the Internet using the average normalization technique to separate the typing samples from the computer platform used to generate them. These keystroke biometric devices can use open-platform tools like Java Applets or CGI programs. Multi-thread Java programming techniques can already be used to capture the user typing rhythm without intrusive impact i.e. not using 'heavy' components as e.g. ActiveX which need special security permissions in order to access to low-level features of the machine. Nowadays, the systems alerts to the user in order to close other applications running during the login process in the website because the program could get varying amounts of processor time, due the counts depend heavily on the other processes on the machine. Future work could be focused on solving this issue.

The presented system is good for the B2C e-commerce model where we want to reach a broad market i.e. it is good because it is a way to increase the security related to the authentication process without requiring special biometric hardware devices like fingerprint readers and so. The prototypes developed in our work have a low cost in hardware and software in comparison to other traditional biometric devices (fingerprint, iris-scan, etc.) and 'brute force' attacks are useless against them because they should generate also the interkey/holdkey times in each password typing (sample).

References

1. Bolle,R., Jain,A., Pankanti,S. *Biometrics. Personal Identification in Networked Society.* Kluwer Academic Publishers. (1999)
2. Gaines,R., Lisowski,W., Press,S., Shapiro,N. Authentication by keystroke timing: some preliminary results. In *Rand Report,* R-256, NSF, Rand Corp., Santa Monica, CA. (1980)
3. Monrose,F., Rubin,A.D., Keystroke dynamics as a biometric for authentication. *Elsevier Science Direct, 16, (no.4):351-359.* (2000)
4. Obaidat,M.S., Sadoun,B. Verification of computer users using keystroke dynamics. *IEEE Trans. on Systems, Man and Cybernetics. Vol.27, No.2*, 261-269. (1997)
5. Tapiador,M., Sigüenza,J. Fuzzy Keystroke Biometrics on web security. In *AutoID'99 Proceedings: workshop on Automated Identification Advances Technologies*, Summit, New Jersey, USA, October 1999, IEEE Robotics and Automation Society, 133-136. (1999)
6. Wayman, J.L. Technical Testing and Evaluation of Biometric Identification Devices. In *Biometrics. Personal Identification in Networked Society.* 345-368. Kluwer Academic Publishers. (1999)

Diffusion Behaviour of Cryptographic Primitives in Feistel Networks

Vasilios Katos

Department of Information Systems and Computer Applications,
University of Portsmouth,
Burnaby Terrace, Portsmouth, PO1 3AE
vasilios.katos@port.ac.uk

Abstract. The concept of product encryption is resident in the majority of symmetric block ciphers. Along with product encryption, two properties were also defined by Shannon, namely diffusion and confusion. In a product cipher such as a Feistel Network (FN), or generally a Substitution Permutation Network (SPN), diffusion is dependent upon two types of primitives, the nonlinear transformation and the swapping scheme. Different approaches to diffusion analysis considered either the topology of a FN, or the nonlinear transformation. This paper describes a metric for diffusion in a way suitable for investigating the behaviour of the underlying primitives of a FN.

1 Introduction

Since their invention, Feistel Networks (FNs) [1], [2] have been extensively studied and analysed [3], [4]. The large research interest in FNs was due to several reasons:

- flexibility of the underlying non-linear primitive. The main non-linear function involved in a FN, which is not required to be injective, in order to allow unambiguous decryption;
- realisation of product encryption. FNs are excellent examples of product encryption. The concept of product encryption, introduced in [5], states that a chain encryption of “weak” ciphers results into a much stronger one. In the same paper, the notion of confusion and diffusion was introduced, which relate to the cryptographic qualities of a cipher;
- the DES [6], which is probably the most analysed cipher, is a FN.

However, the bulk of the research in FNs is on homogeneous balanced FNs [3], since the DES falls into this category. As a direct consequence, the research interest focused on the construction and properties of the underlying non-linear function. In [3] there is an investigation of the topology of a FN rather than the non-linear function. In the same paper, confusion and diffusion were put into perspective and metrics such as the diffusion rate and confusion rate where defined. A similar perspective is in [4], but the methodology for examining the diffusion involved directed graphs. However, although that a graph is an effective tool, the diffusion capability of a cipher is not apparent as the complexity increases.

The contribution of this paper is two-fold. First, it provides a step towards an algebraic description of the diffusion capacity of a FN round. This would allow investigation of a much broader category of FNs, namely the unbalanced heterogeneous FNs. Second, the proposed approach allows assumptions about the non-linear function which can be experimentally evaluated. To demonstrate this, a randomness test is described and can be used for evaluating the behaviour of the FN as a pseudorandom function [7],[8].

2 Diffusion instances and diffusion matrix

The idea behind the construction of the diffusion instances is related to the calculation of the differential characteristic, which is the centrepiece of differential cryptanalysis [9]. A block cipher can be viewed as a function with two independent input variables, namely the plaintext (or ciphertext) and the encrypting (or decrypting) key, and one dependent output variable, the ciphertext (or plaintext).

Diffusion is the property where a given input plaintext bit has the chance to affect the output bits [5]. The higher the diffusion, the more output bits can be affected by a certain input bit. In the described method, the diffusion instance is defined. The diffusion instance is a *snapshot* of the diffusion capacity of a cipher.

The process for generating the diffusion instance is similar to the bitwise calculations used for the Strict Avalanche Criterion (SAC) investigation [10]. Given a random plaintext $p_0 \in_U GF(2)^n$ and a nonzero vector $\alpha = (1\ 0\ 0\ ...\ 0)$, we compute:

$$\psi_j = e_k(p_0) \oplus e_k(p_0 \oplus (\alpha \gg j)),\ 0 \leq j \leq n-1 \tag{1}$$

where $(\alpha \gg j)$ represents the right shift of α by j bits.

If $a[k]$ denotes the k-th bit of the binary string a, then matrix Ψ is defined as:

$$\mathbf{\Psi} = \begin{bmatrix} \psi_1[0] & \psi_1[1] & \dots & \psi_1[n-1] \\ \psi_2[0] & \psi_2[1] & \dots & \psi_2[n-1] \\ \vdots & \vdots & \vdots & \vdots \\ \psi_n[0] & \psi_n[1] & \dots & \psi_n[n-1] \end{bmatrix} . \tag{2}$$

The matrix Ψ would then be one diffusion instance. According to the definitions of the characteristics of confusion and diffusion, for a cipher these characteristics are at maximum if a (binary) swap of any of the input bits results to a swap of the output bits with probability of 0.5 for every output bit. The diffusion instance represents the ability of an input bit to affect an output bit, [11].

The diffusion matrix is calculated from the logical OR of the Ψ matrices:

Definition 1. *Let Ψ_i, $i = 1, 2, ...$ be the diffusion instances of a FN. The diffusion matrix is defined as:*

$$\mathcal{D} = \bigvee_i \Psi_i\ . \tag{3}$$

Theoretically, in order to obtain the actual diffusion matrix of a FN, all plaintexts must be considered. In practice, for a FN with a 64 bit input, it appeared that 10 random plaintexts (and therefore 10 diffusion instances, accounting to a total of 640 plaintexts) would suffice for determining the diffusion matrix. More analytically, after combining 10 diffusion instances, there was no change in the resulting diffusion matrix with each additional diffusion instance. Furthermore, for a block cipher with maximum diffusion capabilities, all entries of its diffusion matrix were equal to one, in the neighbourhood of 10 diffusion instances. Considering a potentially strong block cipher with maximum diffusion capabilities, it is expected that each diffusion instance would include $(1/2) * n$ ones. Therefore, the ith diffusion instance would be expected to contribute with $(1/2)^i * n$ ones in the diffusion matrix. Alternatively, the probability that the calculated diffusion matrix for a potentially strong block cipher is not the actual one, would be $(1/2)^i$. It should also be highlighted that since the key information is not considered, the proposed approach is applicable only on block ciphers where their structure is not dependent on the key.

The diffusion matrix shows if a pairwise relation exists between input and output bits - that is, if a change of a particular input bit has the chance to affect a particular output bit. The diffusion matrix is very helpful in examining product ciphers, because it has the following property:

Lemma 1. *Let C be a FN of j rounds. The diffusion matrix of the cryptosystem is equal to:*

$$\mathcal{D}_C = \beta(\mathcal{D}_1 \cdot \mathcal{D}_2 \cdot \ldots \cdot \mathcal{D}_j) \tag{4}$$

where $\mathcal{D}_i$ is the diffusion matrix of the ith round and $\beta(\cdot) : N \rightarrow \{0, 1\}$ is defined as:

$$\beta(n) = \begin{cases} 1, & \textit{if } n \neq 0 \\ 0, & \textit{if } n = 0 \end{cases} . \tag{5}$$

Proof. The case of a two round FN is shown, that is $\mathcal{D} = \beta(\mathcal{D}_1 \cdot \mathcal{D}_2)$. Let $[\cdot]$ be a boolean evaluation, which evaluates the expression within the brackets to one if it is true and to zero is it is false, such as [p is prime]. The elements of $\mathcal{D}$, $\mathcal{D}_1$ and $\mathcal{D}_2$ are denoted by δ_{ij}, δ'_{ij} and δ''_{ij} respectively. Note that the output of round one is equal to the input of round two. For the first leftmost input bit it is:

$$[\text{input bit 1 is related with round-1 output bit } j] = \delta'_{1j},\ 1 \leq j \leq n \tag{6}$$

from the definition of the diffusion matrix. Similarly, for the first leftmost output bit:

$$[\text{output bit 1 is related with round-2 input bit } j] = \delta''_{j1},\ 1 \leq j \leq n\ . \tag{7}$$

Combining (6) and (7) we obtain:

$$[\text{input bit 1 is related with output bit 1}] = \delta'_{11} \cdot \delta''_{11} + \delta'_{12} \cdot \delta''_{21} + \ldots + \delta'_{1n} \cdot \delta''_{n1} \tag{8}$$

where the right-hand-side is a boolean expression, i.e. $. + .$ denotes the boolean **OR** and $. \cdot .$ denotes the boolean **AND**. If this is repeated for all input and output bits it gives:

$$[\text{input } i \text{ is related with output } j] = \delta_{ij} = \delta'_{i1} \cdot \delta''_{1j} + \delta'_{i2} \cdot \delta''_{2j} + \ldots + \delta'_{in} \cdot \delta''_{nj},\ 1 \leq i, j \leq n$$

or equivalently,

$$\mathcal{D} = \beta(\mathcal{D}_1 \cdot \mathcal{D}_2) \ . \qquad \square$$

From the diffusion matrix, we can calculate the diffusion, which is defined as the ratio of ones:

Definition 2. *The diffusion of a block cipher with a diffusion matrix $\mathcal{D}$ of size $(n \times n)$ is the quantity:*

$$D \triangleq \frac{\#\{\delta_{ij} | \delta_{ij} = 1, 1 \leq i, j \leq n\}}{n^2} \ . \tag{9}$$

Obviously, $D \in [0, 1]$. This definition of diffusion, combined with Lemma 1 can be used for assessing the diffusion of any product block cipher, provided that the diffusion matrices of the underlying rounds are known. We will demonstrate this by applying it onto FNs.

2.1 FN analysis

The diffusion matrix of a one round balanced FN would look like:

$$\mathcal{D} = \begin{bmatrix} \mathbf{O}_{n/2} & \mathbf{I}_{n/2} \\ \mathbf{I}_{n/2} & \mathbf{F} \end{bmatrix} \tag{10}$$

where $\mathbf{O}_{n/2}$ is a zero square submatrix, $\mathbf{I}_{n/2}$ is the identity submatrix and $\mathbf{F}$ is the diffusion matrix of the round function. In a balanced FN, all submatrices are of size $n/2$. The diffusion of this round would be equal to:

$$D_1 = \frac{4n + n^2 D_f}{4n^2} \tag{11}$$

where D_f is the diffusion of the round function. It can bee seen that the diffusion of a one round balanced FN is upper bounded by $(4 + n)/4n$ and therefore it cannot offer complete diffusion. To calculate the diffusion of a two round balanced FN, we apply Lemma 1:

$$\mathcal{D}_2 = \beta(\mathcal{D}_1 \cdot \mathcal{D}_1) = \begin{bmatrix} \mathbf{I}_{n/2} & \mathbf{F} \\ \mathbf{F} & \beta(\mathbf{F} \cdot \mathbf{F}) \end{bmatrix} \tag{12}$$

where it can be seen that the diffusion for a two round balanced FN can be at most $(3n^2 + 2n)/4n^2$. For a three round balanced FN, the diffusion can reach its maximum value, 1.

We observe that no matter how *strong* the round function is, the diffusion of a two round balanced FN is limited by the boundary 3/4. The reason for this is the structure of the diffusion matrix. The permutation of the columns of the matrix is directed by the Swapping Scheme, SS, which appears after the nonlinear transformation in a Feistel round. Although that the SS does influence the diffusion of the FN, it does not actually increase it; the increase is due to the application of the non-linear transformation. Typically, a SS is a permutation of the input bits. In a balanced FN the permutation is the swap between the $n/2$ leftmost bits and the $n/2$ rightmost bits. This swap is responsible for the symmetry in the diffusion matrix. However, each application of SS would not increase the diffusion:

Corollary 1. *The product encryption of a block cipher with diffusion equal to D and a SS, results to a cipher with the same diffusion (D).*

The proof follows from the fact that the diffusion matrix of the SS is a matrix with exactly n nonzero elements, arranged in a way that every row has exactly one nonzero element (i.e. the rank of the matrix is n). The identity SS is an instance of a SS where the diffusion matrix is the identity matrix.

The inherent structure of the FN diffusion matrix reveals the limitations of its diffusion capacity. Since the diffusion D measures the density of ones in the matrix, it follows that $1 - D$ would correspond to the density of zeros. It is therefore desirable that $1 - D$ reaches zero, in order to attain maximum diffusion. As observed above, in a two round FN with the ”traditional” swapping of the left and right input blocks, the number of zeros would be at least $1-(3n^2+2n)/4n^2$, i.e. it would reach asymptotically $1/4$ as n increases.

We now consider a two round Substitution Permutation Network, SPN [2], [12], where each round includes a non-linear function of the same diffusion D_1 as our FN above. For simplicity, it is assumed that these two rounds include different nonlinear functions, although their diffusion is the same, $D_1 = D_2$. We also consider the permutation to be a random SS, i.e. a random permutation of the input bits, rather than a tidy swapping of the left and right input block. The diffusion of the one round instances would be:

$$D_1 = D_2 = \frac{4n + n^2 D_f}{4n^2} \tag{13}$$

where D_f denotes the diffusion of the underlying nonlinear function. However, in a SPN construction it is possible that the zeros are placed randomly in the diffusion matrix. Therefore, the expected zeros in the diffusion matrix of the two round SPN for $D_f = 1$ would be (for the proof see Lemma 2, section 3):

$$(2(1 - D_1) - (1 - D_1)^2)^n = \left(\frac{15n^2 - 56n + 16}{16n^2}\right)^n \tag{14}$$

which is small (< 0.006) for most values of n ($n \geq 6$). From this result the inefficiency of FNs with respect to diffusion is apparent.

As mentioned above, Lemma 1 is useful when analysing the diffusion of product ciphers. For instance, FEAL-4 [13] is a four round FN with the characteristic that the leftmost half input is added (modulo 2) to the rightmost half input, before the first FN round. Considering the product encryption of the first addition and the first round, the diffusion at the end of the first round would be:

$$\beta(\begin{bmatrix} \mathbf{I}_{32} & \mathbf{O}_{32} \\ \mathbf{I}_{32} & \mathbf{I}_{32} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{O}_{32} & \mathbf{I}_{32} \\ \mathbf{I}_{32} & \mathbf{F} \end{bmatrix}) = \begin{bmatrix} \mathbf{O}_{32} & \mathbf{I}_{32} \\ \mathbf{I}_{32} & \mathbf{F} \end{bmatrix} \tag{15}$$

i.e. the additional complexity of the initial addition is completely redundant and unnecessary from a diffusion perspective, since for FEAL $D_f = 1$.

3 The diffusion randomness test

Statistical tests for randomness [14]-[16] are of a particular interest in cryptography, since they are one of the approaches for assessing the cryptographic strength of a cipher. This section describes a randomness test utilising the diffusion instances, $\varPsi$.

For a potentially strong cipher, the number of zeros must be equal to the number of ones in every row of the diffusion instance. Furthermore, for a potentially strong cipher, (statistically) all runs of $\varPsi$ table constructions should result to having the number of ones equal to the number of zeros. However, such an examination does not give any indication about existing linear relations between the elements in the matrices. For instance, if $\psi_2[1] = \psi_3[2]$ with probability different to 0.5, there is a linear relation between input bits 1 and 2 [17].

The diffusion randomness test deals with the similarities of the diffusion instances, $\varPsi$. For a potentially strong cipher the following criteria for the $\varPsi$ matrices are set:

- the number of ones should be equal to the number of zeros,
- the ones (and zeros) should be *randomly* distributed in the matrix,
- $\varPsi_i$ and $\varPsi_j$ should not be *similar* for $i \neq j$.

The first criterion denotes that the cipher is not biased toward ones or zeros. This is inherently related to the confusion of a cipher, where it is desirable that the chance of an output bit inverting is 0.5, given an inversion of an input bit. Published statistical tests for randomness, such as the frequency test [14] can be used.

The second and third criterion include arbitrary terms and need to be quantified. The test described in this paper attempts to provide means for measuring the randomness and similarity of the matrices as follows. The randomness test is based on the following Lemma.

Lemma 2. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two square matrices and* p_a *and* p_b *be the densities of zeros in each matrix respectively. If the zeros are distributed randomly in the matrices, then the* expected *density of zeros in their product* $\mathbf{C} = \mathbf{A} \times \mathbf{B}$ *would be:*

$$p_c = (p_a + p_b - p_a p_b)^n \tag{16}$$

where n *is the dimension of the matrices and the multiplication operation is performed in the set of integers.*

Proof. For $\mathbf{A}$, the density of zeros would be:

$$p_a = P(a_{ik} = 0) = \frac{\#(\text{zeros in } \mathbf{A})}{n^2} \tag{17}$$

Similarly, for B:

$$p_b = P(b_{kj} = 0) = \frac{\#(\text{zeros in } \mathbf{B})}{n^2} \; . \tag{18}$$

For every element in C, the following relation holds:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \; . \tag{19}$$

The probability to obtain a zero is obtained from (19):

$$P(c_{ij} = 0) = \prod_{k=1}^{n} P(a_{ik} = 0 \cup b_{kj} = 0) = (p_a + p_b - p_a p_b)^n \; . \qquad \square$$

By comparing the actual and estimated values, it is tested whether a cryptographic primitive behaves as a random source when generating the $\mathbf{\Psi}$ matrices. That is, in the case of a random source the zeros will be randomly placed in the matrices and there would be no consistent placement whatsoever. We argue that if the actual and estimated values are (statistically) different, then the underlying cryptographic primitive does not yield a pseudorandom function. The opposite is not necessarily true; a primitive passing the test does not imply that it is a pseudorandom function, since the test does not provide any indication about the computational indistinguishability of the primitive [18].

```
diff_rand_test(A,B){
  p_a = zeros_density(A);
  p_b = zeros_density(B);
  p_c = zeros_density(A*B);
  if (abs(p_c-(p_a+p_b-p_a*p_b)^n)>significance_level )
    then return ('fail')
    else return ('pass') }
```

Unfortunately for a relatively large n ($n > 40$) and $p_a, p_b < 2/3$, the density of zeros is negligible for both expected and actual values and therefore the randomness test would not produce significant results. Therefore it is suggested that the $\mathbf{\Psi}$ matrices are partitioned and the test is applied onto the partitions (submatrices). This is particularly applicable in FNs, where there are emerging submatrices due to the non uniformal treatment of input and output bits.

For the case of a balanced FN, the $\mathbf{\Psi}$ matrix would consist of four submatrices $\mathbf{Q}_i$ as follows:

$$\mathbf{\Psi} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{Q}_3 & \mathbf{Q}_4 \end{bmatrix} \tag{20}$$

and the test would then run as: diff_rand_test($\mathbf{Q}_i$,$\mathbf{Q}_j$), where $i \neq j$. It is expected that a three round balanced FN with an underlying round function being a pseudorandom function would pass the test, although that passing the test would not imply that the round function is pseudorandom. Applying this assumption to the well studied DES, it was established that the three round FN with the DES primitive did not pass the test, confirming the validity of the test (Table 1). The fact that DES could not pass the test is a direct consequence of the the inability of DES to reach complete diffusion in three rounds.

Table 1. Significant differences in DES

product	expected	actual	difference	diff_rand_test()
$\mathbf{Q}_1 \times \mathbf{Q}_2$	0.241739	0.216797	2.5	fail
$\mathbf{Q}_1 \times \mathbf{Q}_3$	0.204115	0.179688	2.4	fail
$\mathbf{Q}_1 \times \mathbf{Q}_4$	0.126188	0.077148	4.9	fail

4 Conclusions

Clearly the reason to adopt a FN structure in a block cipher is mainly due to the convenience it offers, such as ease of moving between encryption and decryption, and less due to its diffusion capabilities. High diffusion in a product cipher implies that the input bits are be treated uniformly in every round. Since this is not the case for a FN, additional complexity (e.g. more rounds) would be required. The proposed description and metric of diffusion enables both the investigation of the topology (structure) of a FN as well as the underlying non-linear function(s). This would allow the investigation of FNs consisting of different round functions, with varying input and output lengths as well as different swapping schemes (unbalanced heterogeneous FNs).

Although that the proposed approach initially aimed for studying FNs, most product block ciphers can benefit from such an analysis.

References

1. Feistel, H.: Block Cipher Cryptographic System, U.S. Patent #3,798,359 (1974).
2. Feistel, H., Notz, W. A., Smith, J. L.: Some Cryptographic Techniques for Machine-to-Machine Data Communications. Proceedings of the IEEE (1975) 1545–1554.
3. Schneier, B. and Kelsey, J.: Unbalanced Feistel networks and block cipher design. Proc. Fast Software Encryption, Lecture Notes in Computer Science, vol. 1039, Springer-Verlag (1996) 121–144.
4. Nakahara J. Jr., Vandewalle, J., Preneel, B.: Diffusion Analysis Of Feistel Networks (Extended Version). citeseer.nj.nec.com/article/nakahara99diffusion.html (1999).
5. Shannon, C. E.: Communication Theory of Secrecy Systems. Bell Systems Technical Journal, vol. 27 (1948) 623–656.
6. FIPS PUB 46: Data Encryption Standard. US Department of Commerce/ National Bureau of Standards (1977).
7. Goldreich, O., Goldwasser, S., Micali, S.: How to Construct Random Functions. Proceedings 25th Annual Symposium in Comp. Sci. (1984).
8. Luby, M. and Rackoff, C.: How to Construct Pseudorandom Permutations from Pseudorandom Functions. SIAM J. Computing, vol.17, no.2 (1988) 373–86.
9. Biham, E. and Shamir,A.: Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology. Vol. 4, No. 1 (1991) 3–72.
10. Webster, A. and Tavares, S.: On the design of S-boxes. In H. Williams (ed), Crypto'85, LNCS No. 218, Springer: Berlin Heidelberg New York (1986) 523–534.
11. Pfleeger, C.: Security in Computing. London: Prentice Hall (1989).
12. Heys, H. and Tavares, S.: Substitution Permutation Networks resistant to Differential and Linear cryptanalysis. Journal of Cryptology, no.9, vol. 1 (1996) 1–19.

13. Shimizu, A. and Miyaguchi, S.: Fast data encipherment algorithm FEAL. Advances in Cryptology, Eurocrypt'87, LNCS no.304, Springer: Berling Heidelberg New York (1988) 267–280.
14. Knuth, D.: Seminumerical algorithms. The Art of Computer Programming, vol 2. Addison-Wesley: New York (1981).
15. Rukhin, A., Soto, J., Nechvatal, V., Smid, M., Barker, E., Leigh, S. Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J.: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22 (2000).
16. Beker, H. and Piper, F.: Cipher Systems: The Protection of Communications. Van Nostrand Reinhold (1982).
17. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. Advances in Cryptology EURO-CRYPT '93, LNCS 765 (1994) 386–397.
18. Blum, M. and Micali, S.: How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits. SIAM Journal on Computing, Vol.13 (1984) 850–864.

Authentication and Authorisation for Integrated SIP Services in Heterogeneous Environments

Dorgham Sisalem, Jiri Kuthan

Fraunhofer Institute for Open Communication Systems (FhG Fokus)
Kaiserin-Augusta-Allee 31, 10589 Berlin,
Email:sisalem@fokus.fg.de, kuthan@fokus.fhg.de

Abstract: In order to provide secure and high quality IP-based communication in heterogeneous environments there is a clear need to couple the signalling protocols used for establishing such communication sessions with supporting components and services providing QoS control, security and mediations between different technologies. In this paper we will be investigating the issue of providing an authorization infrastructure for VoIP based sessions that allows the establishment of VoIP sessions and coupling those sessions with a row of supporting services.

1 Introduction

The session initiation protocol (SIP) [1] was primarily designed as a tool for establishing and controlling communication sessions between two or more end systems or users. With this regard, SIP is increasingly being hailed as the standard protocol for VoIP and instant messaging in both the Internet as well as 3G UMTS networks as part of the IP-based multimedia subsystem (IMS).

In a perfect world, having access to an IP network paired with a signalling protocol such as the session initiation protocol (SIP) [1] would be sufficient to establish end-to-end communication between any two users. However, in reality and especially in wireless environments such as UMTS networks, a row of other supporting services is required to transparently establish a communication session between mobile users with an acceptable QoS level. Further, as depicted in Figure 1 various translation and transcoding services are needed to allow the establishment of a communication session in heterogeneous environments. The heterogeneity might be caused by the following factors:

- **End devices:** This includes end devices using different media representation approaches. This involves different compression styles or text or audio capabilities only.
- **Communication protocols:** This involves establishing communication sessions between entities using different protocols for establishing these sessions. This includes establishing a call between a SIP-based device and an ISDN/GSM phone or SIP to H.323.
- **Security policies:** This involves establishing a session between a user in a private IP network and a user in the public Internet for instance.

To overcome this heterogeneity and allow transparent session establishment a row of so-called supporting services is required. These supporting services include the following examples:

- **QoS Establishment Services:**. This indicates mechanisms for providing assured resources in terms of bandwidth for the media sessions established with SIP. Especially in networks with scarce but valuable bandwidth resources such as wireless networks, the session establishment needs to be coupled with the mechanisms that are provided by mobile network operators (MNOs) for ensuring the availability of the needed resources for the session.
- **Connection Services for Heterogeneous Networks:** When contacting users in a non-SIP environment, i.e., users not using SIP as their signalling protocol such as PSTN and GSM users, the SIP signalling needs to be terminated at the one side and translated to the other protocol. Thereby to achieve transparent communication between the users of the two environments a service provider needs to support gateways between these environments.
- **Firewall and Network Address Translation Services:** These services indicate components that are used to protect private networks form attackers as well hide their internal structure. Such components include firewalls and network address translators (NATs). Firewalls usually have a set of fixed rules indicating which ports and addresses can be reached from the outside as well as which addresses and port numbers the users are allowed to connect to from the inside. NATs are used to map a row of private addresses and port numbers to a smaller number of public IP addresses and port numbers. This has effect of hiding the internal addressing structure of the private network and reduces the expenses of buying larger sums of public IP addresses. As SIP users dynamically negotiate addresses and port numbers static firewall rules cannot be used, as the system administrator has no advance knowledge of addresses and port numbers to be used for the communication [2]. Thereby, to allow SIP signalling and media exchange over firewalls and NATs some interaction between the SIP infrastructure and the firewalls and NATs is needed to allow dynamically changing the firewall rules and mirror possible address translations in the SIP messages [3].
- **Media Transcoding and Translation Services:** This type of service can be used to allow users using devices with incompatible compression styles for example to communicate with each other. As a possible supporting service, a service provider might offer translation and transcoding services such as speech to text transcoders to allow a hearing impaired person to contact another person that is using a voice only device.
- **Conference Services:** As a further supporting service, a service provider might offer a conference server for enabling small to medium sized conference sessions. This service might include a media mixer and a centralized conferencing site at which users might login, initiate a session and invite other users to join the conference.

Thereby, providing a SIP-based communication infrastructure implies some sort of integration between the above mentioned services and SIP. This might involve some modification and enhancement of the SIP signalling itself but also a tight correlation in the authentication, authorization and accounting (AAA) procedures. In this paper we will be investigating the issue of providing authentication and authorization mechanisms for SIP based sessions that allow establishing SIP sessions and coupling those sessions with a row of supporting services. In a first step, see Sec. 2, we will briefly describe the common approaches for authenticating a user's identity. The major part of the work, see Sec. 3, will then be dedicated for describing possible approaches for authorizing a user's request for a service consisting not only of the SIP session but also of supporting services. The described mechanisms will then be evaluated in terms of their applicability, scalability and security among other features in Sec. 4.

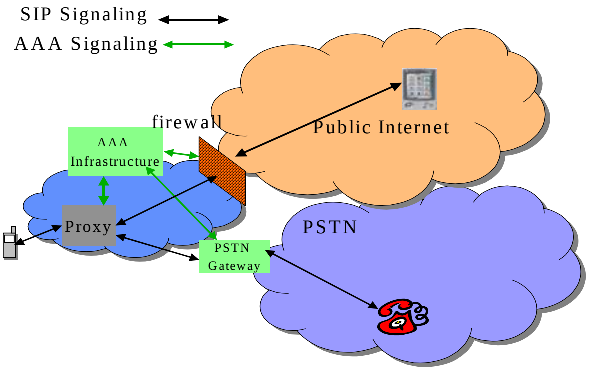


Figure 1. SIP in heterogeneous environments

2 Authenticating Service Requests

The main goal of the authentication procedure is to provide a proof of identity of both the users and providers. For proving the identity of a provider, schemes based on trusted digital certificates are usually used such as with TLS, see [8].

For authenticating users, we can in general distinguish two approaches:

- **Request-based authentication:** With this mechanism the service provider authenticates each request issued by the user. This in general involves a challenge-reply kind of mechanisms such as HTTP Digest, which was specified to be use with SIP.
- **Session-based authentication:** With this approach the authentication procedure is carried out once before the user starts sending any requests. During this phase the user and provider establish a temporary key that can be used to sign and possibly encrypt all requests sent by the user until the termination of the session. UMTS AKA as described in 3GPP [4] present such approaches

For some support services such as QoS for which the user might issue explicit requests as well, similar authentication mechanisms might be used.

3 Coupling SIP Sessions with Supporting Services

When coupling supporting services with a SIP session there are mainly two possibilities for realizing the authentication and authorization actions: SIP dependent and SIP independent authorization. In the SIP dependent scenario, the authorization actions are dealt with as part of the SIP signalling and the information needed for carrying AAA related information are transported as part of the SIP messages. In the SIP independent scenario, the supporting services use their own protocols for carrying out the required AA steps.

3.1 SIP Dependent Authentication and Authorization

In this case the user gets authenticated and authorized to use a supporting service during the session establishment phase using SIP.

3.1.1 User Initiated Services

In this scenario the end user requests explicitly the service. In order to get authorized to use the service the user needs to present some credentials. These credentials are generated during the SIP session establishment and are often called authorization tokens, see [5] and [6] for more details.

Figure 2 shows a simplified message flow in which the user initiates a SIP session and a service such as QoS is coupled with this session.

1. In the first step the user initiates a SIP session by sending an INVITE message indicating that he would like to use QoS resources. This can be indicated through an extension to the session description protocol (SDP), see [9].
2. The proxy might want to check with a AAA server whether the user is eligible for initiating calls with the indicated message content. The AAA server takes its decision based on local policies as well as the user’s profile, which governs which services the user is allowed to utilize. In case the user is not eligible for using the service, the invitation is rejected.
3. In case of a positive reply the INVITE message gets forwarded towards the receiver.
4. The reply to the INVITE message indicates the callee’s media characteristics and QoS preferences.
5. After receiving the callee’s reply, the proxy has the complete information about the IP addresses and port numbers of the communicating end points as well as the media types, compression styles and bandwidth to be used. This information is then used by the AAA server to create an entry for this session. This entry is indexed by an authorization token that identifies the entry as well as the AAA server generating it and is then given to the proxy.
6. The proxy includes the token into the reply and forwards it to the user.
7. The user can issue a service request, e.g. QoS reservation, which includes the authorization token.
8. To authorize the service request, the service control entity, here a QoS router, can use this token to verify the eligibility of the user to request these resources. This is done by contacting the AAA server identified by the token and informing it about the user’s wishes and the token delivered by the user. The token is then

used as a reference for the authorization information generated during the SIP session establishment. The answer of the AAA server is then made based on the comparison of the parameters of the requested service and the values contained in the entry generated during the session setup.

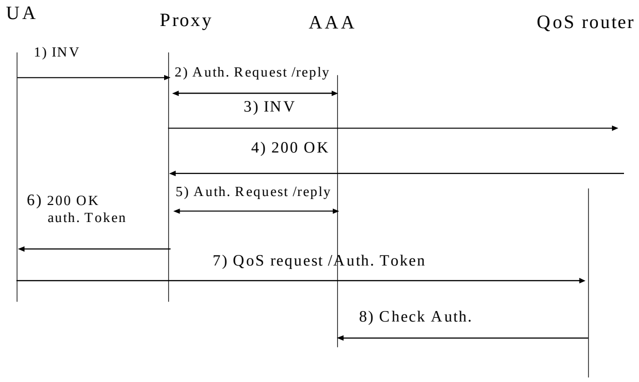


Figure 2. Initiated call in the user initiated services model

In case of session based authentication, the authorization token can be exchanged securely between the SIP proxy either by using an encrypted communication link between the two entities or by encrypting the token using a temporal shared key. The same approach can then be used for exchanging the token between the user and the QoS components. This approach is similar to the one described for 3GPP [4].

In case a mechanisms such as HTTP digest is used for authenticating the user, then the proxy can send the token to the user encrypted with the secret key shared between the user and the SIP provider. The token can then be encrypted with the shared key used between the QoS components and the user for authenticating the user. As the shared keys in this scenario are usually rather short, using tokens in scenarios with request authentication is less secure than for the case of session-based authentication.

3.1.2 Proxy initiated services

In this case the SIP proxy itself initiates the service request and there is no need for exchanging authorization information with the user. This scenario is especially interesting for controlling firewalls and NATs or using a gateway to another network. In this scenario we can distinguish two initiation methods: proxy controlled and proxy routed services.

3.1.2.1 Proxy controlled services

This scenario includes the case for controlling a firewall or a NAT in a midcom like scenario, see [3]. Figure 3 depicts a scenario in which a network is protected by a firewall. This firewall can be controlled by a SIP proxy, which can issue requests to dynamically open certain holes in the firewall and thereby change the filtering rules.

1. After receiving a SIP request the proxy checks with the AAA server whether the user is allowed to make outside calls.
2. If yes, the INVITE gets forwarded to the receiver
3. After receiving the call, the receiver accepts the call and replies with a 200 OK

4. Upon receiving the 200 OK the proxy has the complete information about the addresses and port numbers of the caller and callee. This information is then used to instruct the firewall to change the filtering rules to allow the media traffic of the established session to traverse the firewall.
5. The OK 200 is forwarded
6. Sending an ACK completes the session initiation. Traffic can now flow through the firewall.

Note that in this case no tokens need to be exchanged between the user and the proxy. Thereby both session and request based authentication mechanisms are equal here.

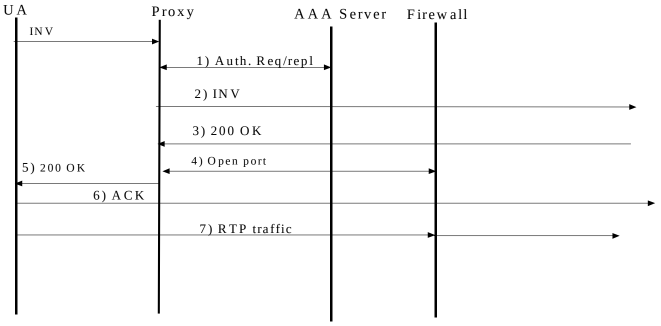


Figure 3 Proxy controlled service

3.1.2.2 Proxy Routed Services

In this scenario, a SIP proxy forwards authenticated and authorized requests to another SIP entity that actually delivers the service. This entity could be a PSTN gateway or some other kind of a transcoding gateway. Figure 4 depicts a scenario in which the user would like to reach a PSTN phone over a gateway.

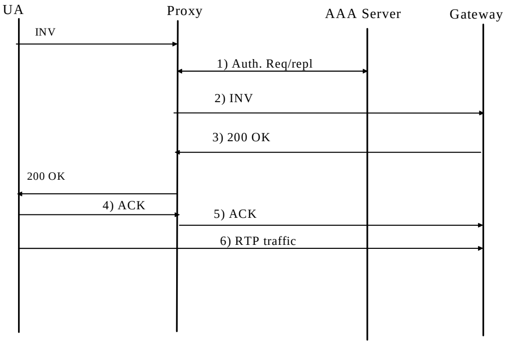


Figure 4 Proxy routed service

1. After receiving a SIP INVITE request for example, the proxy receives the INVITE and checks with the AAA server whether the user is allowed to contact the gateway
2. If the user is authorized to make calls to PSTN destinations the INVITE gets forwarded to the gateway. In this scenario the proxy acts as a kind of a firewall in front of the gateway. Actually it is often the case, that gateways reject all calls not coming from a dedicated proxy. The authenticity of the requests and the assurance that they actually come form a certain trusted proxy, which checks the authorization of the users before forwarding a request, should be guaranteed through a network level security association such as TLS [8] or IPSec between the proxy and the gateway. In order to make sure that all subsequent requests in the session traverse the proxy, the proxy adds a Record-route entry into the INVITE message.
3. The gateway answers with a 200 OK, which is forwarded by the proxy
4. The session establishment is finalized by sending an ACK after which media traffic can be sent to the gateway.

Note that this scenario could also have been realized with the user initiated service scenario, see Sec. 3.1.1. That is the user would receive an authorization token from the proxy and then contact the gateway directly. However, in this case the processing load on the gateway would be increased, as the gateway would need to contact the AAA server to check the correctness of the authorization token.

3.2 SIP Independent Authorization

In this case the user needs to authenticate and authorize himself twice. Once during the SIP session establishment and once during the service request. As depicted in Figure 5 the coupling of the SIP session and the service request is achieved as follows:

1. The user starts the session by issuing a SIP INVITE message.
2. The proxy authenticates and authorizes the user with the help of the AAA server.
3. The INVITE gets forwarded to the destination.
4. The receiver accepts the call by issuing a 200 OK message
5. The OK message gets forwarded to the user.
6. The session establishment is completed by issuing the ACK message.
7. At this stage the user asks for the service.
8. The entity providing the service, e.g., a QoS router, authenticates and authorizes the user. The way this authentication is realized depends very much on the used QoS reservation protocol. For example RSVP proposes the usage of COPs [10] objects, others might use digest authentication similar to SIP.
9. If the user is authorized to use the service then a positive answer is sent.

Notice that the message flow depicted in Figure 5 is only one possibility. As the SIP proxy is not offering expensive services it might not need to authenticate the user at all and thereby we would drop steps 2 and 3. Also, the service request could be established before the SIP session or correlated with it as described in [7].

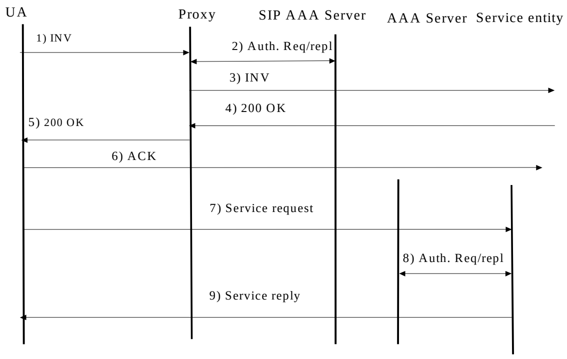


Figure 5. SIP-independent authorization

4 Summary and Conclusions

In this paper we have described various possibilities for enhancing SIP services with a number of supporting services such as QoS, transcoding components and many more. To finalize our work we compare the advantages and disadvantages of the different approaches regarding issues such as performance, security and applicability among others. We will see that choosing the optimal approach for realizing AAA in such a scenario is difficult and depends often on the natures of supporting service.

- **Performance:** In case the user needs to be authorized for both the SIP session and the service usage, the SIP independent approach requires a higher overload in terms of exchanged messages and time. The exact difference depends very much on the authentication mechanisms used by the service entities. For example for the case of the user initiated services and with mechanisms similar to those used for SIP (HTTP DIGEST) we can assume twice the authentication delay and the same time for checking the AAA server. That is in the case of SIP-independent authorization, the service entity would contact the AAA server to check the eligibility of the user. In the case of dependent authorization, the service entity would also need to check the authorization token with the AAA server that generated it. For the case that the SIP session establishment does not require authentication and authorization, both schemes have similar performance. This scenario is especially valid when a user utilizes a public SIP provider which does not require authentication for issuing invitations but still wants to use the QoS infrastructure provided by the network access provider.
- **Applicability:** The applicability of both SIP dependent and independent authorization to the different service scenarios identified in Sec. 2 depends greatly on the service.
  - **QoS establishment service:** Both approaches are applicable to the scenario of coupling QoS reservations with a SIP session. For the case of SIP independent authorization, the QoS protocols need, however, to incorporate

user authentication and authorization more closely with the QoS reservation protocols. This would further increase the complexity of such protocols. With the dependent approach, either the proxies can instruct the QoS components to provide certain QoS features, or the QoS protocols would carry the authorization tokens.

  - **Network security and translation service:** For the case of traversing a firewall, SIP independent authorization does not apply easily as controlling the middle box requires knowledge about all the communicating end systems. NAT traversal is not possible with the SIP independent scenario, as the SIP proxy needs to know the results of the address translation of the media flows already during the signalling phase.
  - **Connection services:** For the case of gateway usage, using the SIP dependent scenario is preferred. The SIP proxy provides a kind of a firewall in front of the gateway filtering unauthorized requests and reducing the load on the gateways that would have been otherwise required to authenticate and authorize the users. The SIP independent scenario is applicable as well but would require the user to authenticate himself directly with the gateway. This would imply, that the gateway needs to maintain its own AAA infrastructure and relation with the user.
- **Security:** This aspect indicates whether the used solution would have negative effects on the security of the communication session or the signalling protocol. Also we need to avoid introducing new possibilities for denial of service attacks or data manipulation
  - For a proxy to authorize a QoS reservation for example, it needs to extract the media description data from the SIP messages and analyze them. This means that SIP messages cannot use end-to-end encryption in the SIP dependent scenario. This is not an issue for the SIP independent scenario.
  - Another aspect is the security of the exchanged authorization token between the proxy and the user in the user initiated scenario. This data usually indicates the entity that generated the token as well as a special entry to the authorization data generated at that entity during the SIP signalling. Stealing this data could allow an interceptor to generate QoS requests under the identity of the actual user involved in the SIP session establishment. As described in Sec. 3.1.1, this can be avoided by encrypting the exchanged tokens. Further, this risk can be reduced by indicating in the AAA entries created during the session establishment phase the exact addresses of the communicating entries. Thereby during the QoS reservation phase only reservations between those addresses can be established. However, this still allows for a denial of service attack. By sending data to the callee and putting the IP address of the caller in the data packets an attacker can reduce the share used by the actual caller of the QoS resources and thereby incur costs on him for resources he did not use. To avoid this case, the communication link between the proxy sending the authorization token and the end systems needs to be secured. This involves establishing a shared key between the involved entities and signing sent packets with this key, which further complicates the session set-up and initial authentication procedures. Another

option would be protect the token so that only the end system that has initiated the session can decipher it. In this case, the SIP provider would encrypt the token using a key shared with the user. The user would decrypt the token and add it to his QoS reservation request. For a better protection, the user might again encrypt the token using a key shared with the QoS provider.

- **Complexity:** The aspect of complexity describes here the changes needed to existing components and the additional overhead required for managing new components.
  - For a proxy to authorize a QoS reservation for example, it needs to not only parse and understand the headers of the SIP message but also the session description part as well. This increases the complexity of the SIP proxy and increases the processing overhead.
  - Another aspect is the authorization part itself. In case the user is authenticated and authorized using SIP then the protocols needed for requesting the supporting services might be simplified and do not require such mechanisms.
  - The token mechanism requires extensions to both SIP as well as the service protocols with headers to include the token. Further, the end user needs to coordinate the usage of both SIP and the supporting service by taking the token from SIP and adding it to the service signalling part.
- **Flexibility:** For supporting SIP-dependant service coupling, there always needs to be some trust relation between the SIP provider and the service provider. This can take the form of a secure connection or might be realized using a trusted AAA infrastructure. Thereby, in order to provide new services, the new service provider needs first to establish this trust relation with the SIP provider. This might lead to delays in the introduction of the service or creating dependencies that might make the entry of new service providers more difficult. With the SIP independent AAA scenario, there is no need for a trust relation between the SIP provider and the service provider. However, this scenario requires trust relations between the user and the service provider.
- **Convenience:** The SIP dependent authorization scenario has the big advantage that the user only needs to establish a trust relation with one provider and is only presented with one bill for the resources he is using. With the SIP independent authorization scenario, the user would need to maintain a contractual relation to the providers of each supporting service he would like to use and establish a new relation for each new service.

References

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Spark, M. Handley, E. Schooler, “Session Initiation Protocol”, RFC3261, June 2002

[2] M. Holdrege and P. Srisuresh "Protocol complications with the IP network address translator (NAT)", RFC 3027, January 2001.

[3] P. Srisuresh, J. Kuthan, J. Rosenberg: “Middlebox Communication Architecture and framework”, February 2001, IETF, Internet Draft

[4] 3GPP Technical Specification 3GPP TS 33.102 V3.6.0: "Technical Specification Group Services and System Aspects; 3G Security; Security Architecture (Release 1999)", 3rd Generation Partnership Project, November 2000

[5] W. Marshall, F. Andreasen, D. Evans, “SIP Extensions for Media Authorization”, Internet draft, May, 2002

[6] Sinnreich, Rawlins, Gross, Thomas, "QoS and AAA Usage with SIP based IP communication", Internet Draft, Internet Engineering Task Force, October 2001

[7]Camarillo, Marshall, Rosenberg, “Integration of Resource Management and SIP”, Internet Draft, April 2002

[8] Dierks, C. Allen, "The TLS Protocol Version 1.0" RFC 2246, January 1999.

[9] M. Handley and V. Jacobson, "SDP: session description protocol," RFC 2327, Internet Engineering Task Force, Apr. 1998

[10] Boyle, J., Cohen, R., Durham, D., Herzog, S., Raja, R. and A. Sastry. "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000

Taxonomy of Trust Relationships in Peer-to-Peer (P2P) Communication

Farookh Khadeer Hussain[1], Elizabeth Chang[1], Tharam Dillon[2]

[1] School of Information Systems
Curtin University of Technology
GPO Box U1987, Perth WA 6845, Australia
{HussainF,ChangE}@cbs.curtin.edu.au

[2] Faculty of IT
University of Technology, Sydney
Broadway, Sydney, NSW 2007, Australia
tharam@it.uts.edu.au

Abstract. Trust between two communicating peers is increasingly catching the attention of the research community. Numerous trust models and trust management protocols have been proposed to enable the task of establishing trust between two communicating peers. In this paper, we enumerate all the possible types of trust relationships between two peers in P2P communication, with examples. Additionally, we discuss the conditions when a given trust relationship between two peers is feasible.

1 Introduction

Trust has been analyzed from social and psychological perspectives [5] and has long been a focal point of interpersonal relationships. Many of us use the word *TRUST* in our daily lives. However, trust has different interpretations and different meanings in different contexts and domains.

In the literature, Marsh was the first person to introduce the concept of trust in computer science [4]. He introduced the notion of trust in distributed artificial intelligence.

In this paper, we focus on the possible trust relationships in peer-to-peer communication. We discuss the conditions under which a given trust relationship is feasible. This paper is organized as follows; Section 1 is brief introduction to trust, in Section 2 we present the various possible trust relationships and Section 3 is the summary of this paper along with future work.

2 Trust Relationships in Peer-to-Peer (P2P) Communication

In this section we discuss with examples the various possible trust relationships in peer-to-peer communication. In Section 2.1, we present a formal definition of what we mean by a trust relationship. In Section 2.2 we discuss the various possible trust relationships. We believe that the various possible trust relationships in P2P communication are:

1. Implicit Trust Relationship
2. Mutual Trust Relationship
3. Group Trust Relationship
4. Federation Trust Relationship

2.1 Definition of Trust Relationship

We define a trust relationship as ‘*a bond or association between the involved peer/s, which signifies the trust between the involved peer/s’*.

In this paper, we use the terms ‘person’ and ‘peer’ interchangeable because a person is always behind a peer and a peer’s behavior depends directly on the person controlling it.

2.2 Implicit Trust Relationship

We define the binding between a person and itself, which signifies the belief in its capability or willingness to perform an action at a given point of time, as ‘Implicit Trust Relationship’. We believe that implicit trust relationships depend on the following factors:

- The capability or willingness of the person to perform a specific action at a given point in time; and
- Whether the person is an optimist or a pessimist.

Some peers have the tendency to believe that they can do everything in this world (optimists), while they may not necessarily be capable of doing it. In contrast, some people may have a pessimistic attitude towards things and they tend to underrate their capabilities (pessimists). We call this relationship of a given peer in itself that signifies the trust that it has in itself to perform a specific action at a given point in time as Implicit Trust Relationship. In implicit trust relationships, only one peer is involved. Implicit trust relationships are a subset of all the rest of the trust relationships.

2.3 Mutual Trust Relationship

We define the binding between a peer and another peer, which signifies the belief in the other peer’s capability or willingness to perform an action at a given point in time, as a ‘Mutual Trust Relationship’. This is the most common form of trust relationships in P2P communication. In mutual trust relationships, there are exactly two peers who are bound by the trust relationship. We believe that implicit trust relationships depend on the following factors:

- The capability or willingness of a peer to perform a specific action at a given point in time , as perceived by another peer;

- The psychological type of the peers involved in the mutual trust relationship [2, 3];
- The outcome of the previous interactions between the peers in the mutual trust relationship; and
- The degree of the trust recommended by the intermediate peer [1].

2.4 Group Trust Relationships

We define a group of peers as '*a collection of more than two peers who perform a set of coherently related tasks and each peer in the group trusts every other peer in the group, for a given context, at a given point in time'*.

We define the binding between a peer and another peer/s, belonging to the same group, which signifies the belief the other peer's capability or willingness to perform an action at a given point in time as 'Group Trust Relationship'.

Unlike mutual trust relationships, group trust relationships can involve two or more than two peers. The peers, however, must belong to the same group which is not the case in mutual trust relationship.

We feel that trust relationship between two peers who belong to the same group is a group trust relationship and not a mutual trust relationship. Additionally, we believe that just like trust, groups are formed for a specific context. Two peers who are members of a group for a specific context may or may not be members of same group for another context.

For example consider the following scenario:

Let us assume that we have a set of peers A1, A2, A3, A4 …..A10. Each peer trusts the other peer for its authenticity at a given point in time. This set of peers A1……A10 is said to form a group where every peer in the group trusts the authenticity of the other members of the group. The trust relationship between A1 and A3 is an example of group trust relationship and not mutual trust relationship.

If A11, which is a new peer, joins this group and all the members in the group trust A11 for its authenticity, then A11 becomes a member of the group for that context.

2.5 Federation Trust Relationships

For defining 'Federated Trust Relationships', we need to first define and explain what we mean by Federated P2P Communication. We define Federated P2P Communication as the communication that takes place between two or more than two groups of peers.

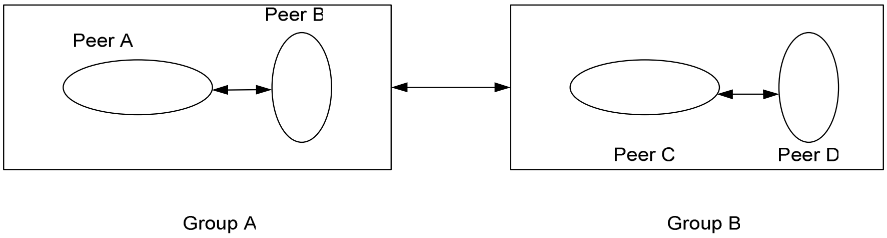
As we mentioned in Section 2.4, the members in each of these groups are centered on a set of coherent interest/s and the peers belonging to a group perform a specific task. In Federated P2P Communication, a peer acts on behalf of the group to which it belongs. If two peers are communicating with each other, it is analogous to two distinct groups communicating with each other. The communication between the groups of peers is regarded as Federated P2P Communication.

We define ‘Federated Trust Relationships’ as the binding between two or more than two distinct groups of peers which signify the peer group’s belief in the peer group’s capability or willingness to perform an action at a given point in time.

A peer in a federated P2P communication may be a member of more than one group. We strongly believe that this is the way that P2P communication will be organized in the future.

For example, let us consider a peer-to-peer file-sharing application, Gnutella. Users of Gnutella can share files with each other. Users may share files like music files, educational documents, documents related to politics etc. Each of these types of files forms a different domain of interest. These domains of interest can be further subdivided. For example, some users who are members of the group which shares music files may be interested in sharing just English songs and not songs composed in any other language. Similarly, some users who are members of the group that shares political documents may be interested in documents related to a particular country/group. A group, as explained above, is formed by a set of peers who have a coherent interest. A peer can join and leave the group at will. Groups can be formed dynamically and destroyed dynamically. Peers can join any group and leave any group.

Let us assume that there are two groups Group A and Group B as shown below. Let us further assume that Peer A and Peer B belong to Group A and Peer C and Peer D belong to Group B.



Let us assume that Peer A who belongs to Group A, wants to interact with Peer C of Group B. The binding between Peer A and Peer C, an example of the federated trust relationships and the communication between them is an example of federated P2P communication

The binding between Peer A and Peer B or the binding between Peer C and Peer D is an example of group trust relationships.

The proposed federated P2P communication structure has all the features that should be present for a P2P communication. We believe that in the future P2P communication will be organized in this way, because a given document can be found with much ease, with less usage of bandwidth as can be found in non-federated P2P communication. In non-federated P2P communication, to locate a particular document, the search query is broadcast to the whole network thus resulting in inefficient use of bandwidth. In federated P2P communication, the query is broadcast only to the members of the group thus leading to a far more efficient use of bandwidth. If a peer is a member of more than one group, the query is broadcast to the group, which has the maximum probability of answering the query. We believe that federation trust relationships may exist between two, or more than two, groups of peers.

References

1. Hussain, F.K., Chang, E. & Dillon,T.S., 'A methodology for reputation management in peer-to-peer (P2P) communication'. (Working Paper)
2. Hussain, F.K., Chang, E. & Dillon, T.S., 2004, 'Factors of trust that affect trustworthiness in peer-to-peer (P2P) based e-commerce', *Proceedings of the International Conference of Business and Information*, Taiwan.
3. Hussain, F.K., Chang, E. & Dillon, T.S., 2004, 'Classification of trust in peer-to-peer (P2P) communication', *International Journal of Engineering Intelligent Systems*, vol. 12.
4. Marsh, S., 1994, 'Formalizing trust as a computational concept', University of Stirling, UK.
5. Smith, J.H., 2002, 'The architectures of trust', Faculty of Humanities, Copenhagen, University of Copenhagen.

A Secure Prepaid Wireless Micropayment Protocol

Supakorn Kungpisdan[1], Bala Srinivasan[2], and Phu Dung Le[3]

[1] School of Network Computing, Monash University
McMahons Road, Frankston, Victoria 3199, Australia
supakorn@sng.its.monash.edu.au
[2] School of Computer Science and Software Engineering, Monash University
900 Dandenong Road, Caulfield East, Victoria 3145, Australia
Bala.Srinivasan@infotech.monash.edu.au
[3] School of Network Computing, Monash University
McMahons Road, Frankston, Victoria 3199, Australia
Phu.Dung.Le@infotech.monash.edu.au

Abstract. In this paper, a secure prepaid micropayment protocol which is suitable for wireless networks is introduced. The proposed protocol employs a secure cryptographic technique that reduces all parties' computation and satisfies transaction security properties, including non-repudiation. This offers the ability to resolve disputes among parties. Compared to existing micropayment protocols, all parties' secret information are well-protected. Finally, we perform an analysis to demonstrate that the proposed protocol has better performance than existing micropayment protocols. As a result, the proposed protocol can be well-operated on limited capability wireless devices.

Keywords. Micropayment, mobile payment, mobile commerce, payment protocol, electronic commerce,

1 Introduction

Micropayments seem to be more widely accepted than other kinds of payments systems for wireless networks because of their lightweight, lower setup cost,and lower transaction cost. Moreover, most payment-related applications for wireless networks are conducted with small-valued goods or services e.g. downloading ring tones, operator logos, or electronic document.

Traditionally, micropayment protocols employ public-key operations and a chain of hash values such as PayWord [6] or NetCard [2]. Although these protocols work well for fixed networks, they are not suitable for applying to wireless networks due to a number of limitations of wireless environments [3, 7].

Recently, a prepaid micropayment protocol called PayFair [8] offers the ability to perform payment transactions on limited computational capability devices. It employs symmetric-key operations and keyed hash functions which reduce the computation at all engaging parties. However, PayFair lacks of transaction privacy since payment information of engaging parties is sent in cleartext during transactions. Moreover, a message sent from a client to a merchant in PayFair lacks of non-repudiation property. Furthermore, a bank is able to impersonate as its clients to perform transactions. In addition, a payment token authorized by the bank is merchant-specific in that it is still can be used to generate the coins to spend with only one specified merchant. Thus, the client is required to request the bank to issue a new payment token every time she wants to perform a payment transaction to a new merchant.

In this paper, we propose a prepaid micropayment protocol which employs a secure symmetric cryptographic technique that not only the computation at all parties, especially at the client, is reduced, but the proposed protocol also satisfies transaction security properties including non-repudiation [1]. Moreover, it offers the ability to resolve disputes among parties. Furthermore, all parties' private information such as payment information and secret keys are well-protected.

In any prepaid payment system, a client has to purchase an electronic coupon which contains spending credits and the amount paid by the client is transferred to a specified merchant before a transaction. In our proposed protocol, we present an efficient method to refund either un-spending credits or coupons. This offers the practicability to the system. Moreover, the coupon in our protocol is general-purposed in that it can be split into smaller value merchant-specific coupons to spend with many merchants.

We analyze the performance of the proposed protocol and compare with PayWord [6] and PayFair [8]. The results show that our protocol has better performance than others in terms of party's computation and the numbers of message passes. Therefore, the proposed protocol can be implemented in limited capability wireless devices with higher performance than existing micropayment protocols.

Section 2 provides overviews of PayWord and PayFair protocols. Section 3 introduces our proposed protocol. Section 4 discusses about security and performance of the proposed protocol. Section 5 concludes our work.

2 Overviews of Existing Micropayment Protocols

In this section, we outline two existing micropayment protocols: PayWord [6] and PayFair [8]. In section 2.1, PayWord is presented to provide an idea about how a micropayment protocol with public-key operations works. In section 2.2, PayFair is outlined to show how to secure transactions using symmetric-key operations.

2.1 PayWord

PayWord [6] is a postpaid micropayment protocol based on public-key cryptography. Three parties are involved in the system: *client*, *merchant*, and *bank*. The client and the merchant establish accounts with the bank. At the beginning of the protocol, the bank issues the client a *PayWord certificate* which contains authorized amount *CL* that the client is allowed to make a payment to each merchant. To make a payment to a merchant, the client generates a set of coins $c_0, ..., c_n$, where *n* = *CL*. The set of c_i is generated as follows: $c_i = h(c_{i+1})$, where $i = 1, ..., n-1$.

In the first payment, the client sends the merchant a *commitment*, which contains the PayWord certificate and c_0, digitally signed by the client. Later on, in each payment, the client sends the coin c_i to the merchant. The merchant can infer the value of the coin by applying a number of hash functions to c_i. At the end of the day, the merchant sends the highest value of c_i together with the commitment to the bank. The bank then deducts the money from the client's account and transfers the money to the merchant's account.

However, PayWord is not suitable for applying to wireless environments because it has high client's computation due to public-key operations. Moreover, a certificate verification process leads to additional communication passes [3]. In addition, payment information, c_0 and c_i, is readable by any party who holds the client's public key. Thus, any party is able trace the client's spending.

2.2 PayFair

PayFair [8] is a prepaid micropayment protocol which employs symmetric-key operations and hash functions. The details of PayFair are shown as follows:

Phase A: Prepaid Phase

C → **B** : $ID_C, O_C, h(O_C, K_C)$ (a)

B → **C** : $\{\{N, RN\}_{SK}, RT\}_{K_C}, N, h(\{N, RN\}_{SK}, N, O_C, K_C)$ (b)

Where *SK* is the secret known only to the bank. K_C is shared between the client and the bank. The client requests the bank by sending order number O_C containing the requested amount. The bank returns the message containing a payment token $\{N, RN\}_{SK}$, which is later used to generate coins. *RN* is a random number generated from the serial number *N* and the secret SK_{RN} known only by the bank. The client generates a set of coins $w_i, i = 0, ..., n$, where $w_n = \{N, RN\}_{SK}$, from the process: $w_i = h(w_{i+1})$.

Phase B: Micropayment Phase

C → **M** : $w_0, N, h(w_0, ID_M, K_C)$ (c)

M → **B** : $w_0, N, ID_C, R_M, h(w_0, ID_M, K_C)$ (d)

B → **M** : $w_0, ID_C, ID_M, YES, h(w_0, ID_C, K_M, R_M, YES)$ (e)

The client sends the message *(c)* containing w_0 to the merchant. The merchant then forwards $h(w_0, ID_M, K_C)$ with relevant information to the bank in *(d)*. After receiving the message, the bank can generate w_n from w_0, *N*, and its own *RN* and *SK*. It then transfers the amount *n* to the merchant's account and sends the response to the merchant in *(e)*. The client can start a payment transaction with the merchant as follows:

C → **M** : $w_i \quad where \quad i = 1, ..., n$ (f)

However, in PayFair, the problem about revealing payment information occurred in PayWord still exists since, in the messages *(c)* and *(f)*, w_0 and w_i are sent in cleartext. In addition, although Yen [8] claimed that payment token w_n is general-purposed, it is still merchant-specific when used, that is, although the coins is merchant-independently generated, they are still can be used to pay only one specific merchant. Thus, the client needs to request the bank for a new payment token every time she wants to make a payment to a new merchant. Moreover, in *(c)*, the bank can impersonate as the client to perform transactions with the merchant.

3 The Proposed Protocol

3.1 Overview of the Proposed Protocol

There are three parties involved in our protocol: *client*, *merchant*, and *bank*. At the beginning of the protocol, a client requests a bank for an authorization to perform micropayment transactions. The bank checks the validity of the client's account and issue a *Bank Coupon* containing the amount requested by the client.

To make payment to a merchant, the client generates a *Merchant Coupon* containing the value specified to the merchant. The value of the merchant coupon must not exceed the value of the bank coupon. This coupon has to be validated by the bank. To validate the merchant coupon, the client generates a set of coins, attaches them into the merchant coupon, and sends to the bank. After the validation, the bank transfers the money with the requested value from the client's account to the merchant's account. The client then can make payments to the merchant up to the amount specified in the merchant coupon. In our protocol, a bank is trusted by its clients to generate correct numbers and values of coins for coin validation purpose, but it is not trusted to create payment initialization requests to merchants by itself. This is because the bank itself can generate the sets of coins. It is possible to generate fake requests on behalf of its clients.

Our proposed protocol is composed of 6 sub-protocols: *Setup, Payment Initialization, Payment*, *Extra Credit Request*, *Coupon Cancellation,* and *Coin Return* protocols. Section 3.2-3.7 demonstrate the details of the protocols.

3.2 Setup Protocol

A client **C** requests a bank **B** for an authorization on making a micropayment transaction with the amount CL_T as follows:

$$\mathbf{C} \rightarrow \mathbf{B}: \quad ID_C, CL_T, T_{CP}, h(CL_T, T_{CP}, Y) \tag{1}$$

Note that CL_T stands for total credits that the client is allowed to spend in the system. T_{CP} is the timestamp when generating the request. $h(CL_T, T_{CP}, Y)$ is used to protect the integrity of the message. The bank checks the validity of the client's account and then deducts the amount CL_T from the client's account. Bank then sends the client a *Bank Coupon* that can be used to perform transactions as follows:

$$\mathbf{B} \rightarrow \mathbf{C}: \quad \{CL_T, T_T, T_{CP}, SN, c\}_Y \tag{2}$$

The bank coupon has unique serial number *SN* assigned by the bank and contains authorized credits CL_T. T_T stands for timestamp when issuing CL_T, and *c* is a random number generated by the bank used for generating coins. With this bank coupon, the client can make payments to many merchants repeatedly up to CL_T. After running out of the credits, the client needs to run this protocol to request the bank for a new CL_T again.

3.3 Payment Initialization Protocol

To make a payment to a merchant **M**, the client generates a set of coins $c_i, i = 0, ..., n$, where $n = CL_T$, as follows:

$$c_n = \{c, T_G\}$$
$$c_i = h(c_{i+1}) \quad where \quad i = 0, ..., n-1$$

The client specifies the amount CL_M to spend with the merchant. The client attaches the coins and CL_M into a *Merchant Coupon*, and sends it to the bank:

$$\mathbf{C} \rightarrow \mathbf{B}: \quad h(c_0, T_G, CL_M, X), h(ID_M, c_0, T_G, CL_M, CL_T, T_T, SN, Y), T_G \qquad (3)$$

Where T_G stands for the timestamp when generating a set of coins $c_0, ..., c_n$. Note that the client can either spend the whole credits to only one merchant or spend some credits to a merchant and spend the rest to other merchants. We can see that *h(*c_0*,* T_G*,* CL_M*, X)* is the payment request from the client to the merchant which is unreadable by the bank. The bank retrieves CL_T and CL_M from h($ID_M, c_0, T_G, CL_M, CL_T, T_T$, *SN, Y)* and checks whether $CL_T < CL_M$. If so, it rejects the request. If $CL_T > CL_M$, the bank calculates the client's remaining credits CL_{TR}, where $CL_{TR} = CL_T - CL_M$. It then maintains the table of CL_{TR} to prevent over-spending problem. At this stage, the bank transfers CL_M to the merchant's account. Then the bank sends the following messages to the client and the merchant:

$$\mathbf{B} \rightarrow \mathbf{M}: \quad \{c_0, T_G, SN, CL_M, h(ID_M, SN, CL_{TR}, T_{TR}, Y)\}_Z, \; h(c_0, T_G, CL_M, X) \qquad (4)$$
$$\mathbf{B} \rightarrow \mathbf{C}: \quad h(ID_M, SN, CL_{TR}, T_{TR}, Y), T_{TR} \qquad (5)$$

Where T_{TR} stands for timestamp when the bank updates CL_{TR}. Note that T_T is updated to T_{TR} after calculating CL_{TR}. The merchant retrieves c_0 and CL_M from the encrypted message. She knows that the client has requested to make the payment to her from $h(c_0, T_G, CL_M, X)$, and the client's request has been authorized by the bank from the message encrypted with *Z* shared between the bank and herself. After receiving the message (5), later on, the client can use $\{CL_{TR}, T_{TR}\}$ to make payment to another merchant.

3.4 Payment Protocol

After completing payment initialization, the client can start the payment to the merchant by sending the coin as follows.

$$\mathbf{C} \rightarrow \mathbf{M}: \quad c_j \qquad (where \quad j = 1, ..., n) \qquad (6)$$

The merchant verifies the requested amount by comparing with c_0. After the verification, she provides goods or services to the client. After each payment, CL_M is deducted. The client is allowed to make the payments up to CL_M without any payment authorization from the bank. If the remaining credits are not enough to make another payment, the client can request the bank for extra credits by running *Extra Credit Request Protocol*.

3.5 Extra Credit Request Protocol

Normally, when a client spends the credits up to CL_M, she needs to run *Setup Protocol* to issue a new bank coupon. In our protocol, we reduce the frequency of doing this process by running *Extra Credit Request (ECR) Protocol* instead. With *ECR Protocol*, the numbers of message passes are reduced. Before the next payment, the client checks whether $j > CL_M$. If so, she still can purchase the goods but she needs to request for extra credits from the bank. The client realizes that, if her request has been approved, her total credits CL_{TR} will be deducted by CL_M. To request for extra credits, the client sends the following message:

$$\mathbf{B} \rightarrow \mathbf{M}: \quad c_j, CL_M, h(ID_M, CL_M, T_G, SN, CL_{TR}, T_{TR}, Y) \tag{7}$$

At this stage, CL_M stands for new credits to spend with specified merchant. The merchant retrieves CL_M and forwards the following message to the bank:

$$\mathbf{M} \rightarrow \mathbf{B}: \quad ID_M, h(ID_M, CL_M, T_G, SN, CL_{TR}, T_{TR}, Y) \tag{8}$$

The bank retrieves CL_{TR} and CL_M, and then calculates a new CL_{TR}, where $newCL_{TR} = currentCL_{TR} - CL_M$. The bank transfers CL_M to the merchant's account, and then sends the response to the merchant as follows:

$$\begin{aligned} \mathbf{B} \rightarrow \mathbf{M}: \quad & h(ID_M, SN, CL_{TR}, T_{TR}, Y), T_{TR}, YES, h(YES, CL_M, T_{TR}, Z) \\ & \text{if approved} \\ & (\text{ or } \textit{Rejected} \quad \text{if client has not enough credits}) \end{aligned} \tag{9}$$

The merchant checks whether the authorized CL_M in $h(YES, CL_M, T_{TR}, Z)$ is equal to CL_M received from the client in *(7)*. If so, the merchant sends the client the following message:

$$\mathbf{M} \rightarrow \mathbf{C}: \quad h(ID_M, SN, CL_{TR}, T_{TR}, Y), T_{TR} \tag{10}$$

The client expects to receive the updated CL_{TR}, where $updatedCL_{TR} = current\, CL_{TR} - CL_M$. She calculates CL_{TR} and compares with the received CL_{TR}. If they are matched, the client can infer the updated bank coupon from CL_{TR}. The above message is considered as a notification of the client's remaining total credits. Note that, to make the payment to a new merchant, the client repeats *Payment Initialization Protocol* with the updated bank coupon without running *Setup Protocol* as that in existing protocols.

3.6 Coupon Cancellation Protocol

In our protocol, a client is able to refund an un-used bank coupon previously purchased from a bank by sending the following message to the bank:

$$\mathbf{C} \rightarrow \mathbf{B}: \quad SN, T_{CR}, h(SN, CL_T, T_T, T_{CR}, Y) \tag{11}$$

Where T_{CR} is timestamp when requesting for coupon cancellation. The bank removes the coupon with the serial number *SN* from its database. This coupon will be no longer used in the system. The bank transfers the amount CL_T to the client's account and sends the response of the client's request to the client as follows:

$$\mathbf{B} \rightarrow \mathbf{C}: \quad CancelOK, (CancelOK, SN, T_{CR}, Y) \tag{12}$$

3.7 Coin Return Protocol

In some situation, a client may want to end transaction with a merchant after spending some credits and request merchant to return her the un-spending credits. This process can be done in the proposed protocol as follows:

$$\mathbf{C} \rightarrow \mathbf{M}: \quad c_{j_{max}}, T_G, h(ID_M, c_0, T_G, Y) \tag{13}$$

Where $c_{j_{max}}$ is the highest-value coins currently spent to the merchant. The merchant checks whether the received $c_{j_{max}}$ is equal to $c_{j_{max}}$ that she has. If they are matched, the merchant forwards the following message to the bank:

$$\mathbf{M} \rightarrow \mathbf{B}: \quad ID_M, c_{j_{max}}, T_G, h(ID_M, c_0, T_G, Y) \tag{14}$$

The bank retrieves $c_{j_{max}}$ and c_0 and calculates returned amount, where $returned\ Amount = CL_M - j_{max}$. Bank then transfers the returned amount to the client's account and updates the client's bank coupon with the new CL_{TR}, where $updatedCL_{TR} = currentCL_{TR} + returnedAmount$. The bank updates the entry in the list at the record containing T_G and c_0, and then sends the acknowledgement to the merchant.

$$\begin{aligned}\mathbf{B} \rightarrow \mathbf{M}: \quad & h(returnedAmount, ID_M, c_0, T_G, CL_{TR}, T_{TR}, Y), \\ & h(returnedAmount, ID_C, c_0, T_G, T_{TR}, Z), T_{TR}\end{aligned} \tag{15}$$

The merchant is notified that the returned amount has been withdrawn and transferred to the client's account from $h(returnedAmount, ID_C, c_0, T_G, T_{TR}, Z)$. Also, she is notified that the set of coins starting with c_0 is no longer valid. The merchant then sends the following message to the client.

$$\mathbf{M} \rightarrow \mathbf{C}: \quad h(returnedAmount, c_0, T_G, CL_{TR}, T_{TR}, Y), T_{TR} \tag{16}$$

The client expects to receive the updated CL_{TR}, where $updatedCL_{TR} = current\ CL_T + returnedAmount$, and $returnedAmount = CL_M - j_{max}$. The client compares CL_{TR} with the received one. If they are matched, she can infer the updated

CL_{TR}. Later on, the client can use the bank coupon with the updated CL_{TR} to make a payment to another merchant.

4 Discussions

4.1 Transaction Security Properties

In this section, we show that the simple cryptographic technique applied to our proposed protocol satisfies the above transaction security properties. The following message demonstrates how the technique works:

$$\textbf{B} \rightarrow \textbf{M}: \quad \{c_0, TG, SN, CL_M, h(ID_M, SN, CL_{TR}, T_{TR}, Y)\}_Z, \\ h(c_0, T_G, CL_M, X) \qquad (4)$$

We can see that all transaction security properties for payment systems [1, 5] are satisfied as follows:

1. ***Party authentication*** is ensured by symmetric encryption and *Y* shared between the client and the bank. The encryption ensures that either the bank or the merchant has originated the message, and *Y* ensures that the bank is the originator of the message.
2. ***Transaction privacy*** is guaranteed by symmetric encryption.
3. ***Transaction integrity*** is guaranteed by $h(c_0, T_G, CL_M, X)$ forwarded from the client.
4. ***Non-repudiation of transactions*** is ensured by $h(ID_M, SN, CL_{TR}, T_{TR}, Y)$ in that the bank cannot deny that it did not generate $\{c_0, T_G, SN, CL_M, h(ID_M, SN, CL_{TR}, T_{TR}, Y)\}_Z$ since it is the only party that holds both *Z* and *Y*.

4.2 Dispute Resolution

Our proposed protocol provides offers the ability to resolve disputes among engaging parties in both direct and indirect manners. According to direct dispute resolution, consider the message (5) in *Payment Initialization Protocol*, we can prove that bank is the originator of this message since $h(ID_M, SN, CL_{TR}, T_{TR}, Y)$ can be retrieved by only the client and the bank, but the client does not have the secret *Z*. Thus, the client is not the originator of the message. However, some messages provide indirect dispute resolution. Consider the message (10) sent from the merchant to the client in *Extra Credit Request Protocol*, although the client can generate this message by herself, she cannot modify the content of the message since it will be later detected by the bank.

4.3 Private Information

In any payment system, the information that is known only by relevant parties such as secret keys, bank account information, price, or goods descriptions is considered as *Private Information* [4]. Revealing such information offers the opportunity to perform various kinds of attacks or to trace the client's spending behavior.

In our proposed protocol, c_0 and $c_{j_{max}}$ are sent in encrypted forms compared to signed messages in PayWord and cleartext in PayFair. Moreover, only c_j is sent from the client to the bank over the air. The bank can infer c_0 from $c_0 = h(c, T_G)$, where *n* stands for the current CL_{TR} and later sends c_0 to the merchant in the message (4). Therefore, the secrecy of the requested amount is preserved.

4.4 Performance Analysis

To demonstrate the practicability of the proposed protocol, we compare our protocol with PayWord [6] and PayFair [8] in terms of performance by focusing on the computation and the numbers of message passes of engaging parties.

Considering the party's computation, we mainly focus on the numbers of cryptographic operations applied to engaging parties. Table 1 demonstrates the numbers of cryptographic operations applied to our protocol, PayWord, and PayFair, respectively. Note that n stands for the computations for generating a set of coins.

Table 1. The number of cryptographic operations of SET, iKP, and KSL protocol at client, merchant, and payment gateway, respectively

Cryptographic Operations		Our Protocol	PayWord	PayFair
1. Signature	C	-	1	-
	M	-	-	-
	B	-	1	-
2. Signature verifications	C	-	1	-
	M	-	2	-
	B	-	1	-
3. Symmetric operations	C	1	-	1
	M	1	-	-
	B	2	-	2
4. Hash functions	C	n	n	n
	M	n	n	n
	B	n	n	n
5. Keyed-hash functions	C	4	-	3
	M	1	-	4
	B	3	-	5

From Table 1, we can see that in our protocol, only symmetric-key operations and hash functions are applied, compared to public-key operations in PayWord [6]. It infers that our protocol has better performance than PayWord. Compared to PayFair [8], the proposed protocol also has less party's computation. Moreover, in PayFair, a client is required to contact a bank for issuing a new coupon and generate a new set of coins every time she runs out of credits whereas the coupon in our proposed protocol is issued only once and can be used to make payments with many merchants. This greatly reduces the computational load at the client. These features result in better performance than PayFair.

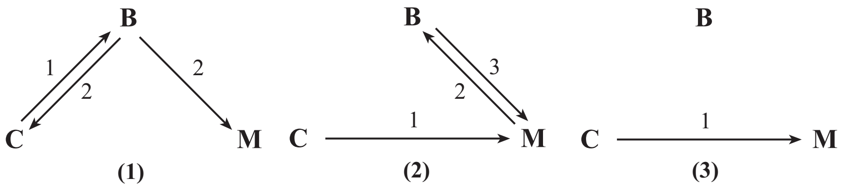


Fig. 1. The numbers of message passes in *Payment Initialization Protocol* of (1) the proposed protocol, (2) PayFair, and (3) PayWord

According to the numbers of message passes, from Fig.1, we can see that the proposed protocol has less numbers of message passes than PayFair which infers better performance. Compared to PayWord, the proposed protocol has higher numbers of message passes. However, PayWord is operated in postpaid mode which a client does not require any payment authorization from a bank in *Payment Initialization Protocol*.

5 Conclusion

We pointed out the problems of existing micropayment protocols when applied to wireless environments due to poor performance and security flaws. We then proposed a prepaid micropayment protocol for wireless networks which solves the above problems. We applied symmetric cryptographic technique which not only reduces parties' computation, but also satisfies transaction security properties. We also performed performance analysis to show that our protocol has better performance than PayWord [6] and PayFair [8] which results in more applicable to limited capability wireless devices.

As our future works, we aim to extend the proposed protocol to perform postpaid micropayments and compare the its results with existing postpaid micropayment protocols including PayWord [6].

References

1. Ahuja, V.: Secure Commerce on the Internet. Academic Press (1996)
2. Anderson, R., Manifavas, C., Sutherland, C.: NetCard - A Practical Electronic Cash System. Lecture Notes in Computer Science, Vol. 1189 (1995)
3. Kungpisdan, S., Srinivasan, B., Le, P.D.: Lightweight Mobile Credit-Card Payment Protocol. Lecture Notes in Computer Science, Vol. 2904 (2003) 295–308
4. Kungpisdan, S., Permpoontanalarp, Y.: Practical Reasoning about Accountability in Electronic Commerce Protocols. Lecture Notes in Computer Science, Vol. 2288 (2002)268–284
5. Park, D.G., Boyd, C., Dawson, E.: Micropayments for Wireless Communications. Lecture Notes in Computer Science, Vol. 2015 (2000) 192–205
6. Rivest, R., Shamir, A.: PayWord and MicroMint: Two Simple Micropayment Schemes. Cryptobytes, Vol. 2(1) (1996) 7–11
7. Romao, A., da Silva, M.: An Agent-based Secure Internet Payment Systems. Lecture Notes in Computer Science, Vol. 1402 (1998) 80–93
8. Yen, S.M.: PayFair: A Prepaid Internet Micropayment Scheme Ensuring Customer Fairness. IEE Computers and Digital Techniques, Vol. 148(6), (2001) 207–213

A Practical Implementation of a Real-time Intrusion Prevention System for Commercial Enterprise Databases

Ulf T. Mattsson

Chief Technology Officer
Protegrity
ulf.mattsson@protegrity.se http://www.protegrity.com

Abstract. Modern intrusion detection systems are comprised of three basically different approaches, host based, network based, and a third relatively recent addition called procedural based detection. The first two have been extremely popular in the commercial market for a number of years now because they are relatively simple to use, understand and maintain. However, they fall prey to a number of shortcomings such as scaling with increased traffic requirements, use of complex and false positive prone signature databases, and their inability to detect novel intrusive attempts. This intrusion detection system interacts with the access control system to deny further access when detection occurs and represent a practical implementation addressing these and other concerns. This paper presents an overview of our work in creating a practical database intrusion detection system. Based on many years of Database Security Research, the proposed solution detects a wide range of specific and general forms of misuse, provides detailed reports, and has a low false-alarm rate. Traditional commercial implementations of database security mechanisms are very limited in defending successful data attacks. Authorized but malicious transactions can make a database useless by impairing its integrity and availability. The proposed solution offers the ability to detect misuse and subversion through the direct monitoring of database operations inside the database host, providing an important complement to host-based and network-based surveillance. Suites of the proposed solution may be deployed throughout a network, and their alarms man-aged, correlated, and acted on by remote or local subscribing security services, thus helping to address issues of decentralized management.

Key-Words. Isolation, Intrusion Tolerance, Database Security, Encryption, GLBA, HIPAA.

1 Introduction

Most companies solely implement perimeter-based security solutions, even though the greatest threats are from internal sources. Additionally, companies implement network-based security solutions that are designed to protect network resources, despite the fact that the information is more often the target of the attack. Recent development in information-based security solutions addresses a defense-in-depth strategy and is independent of the platform or the database that it protects. As organizations continue to move towards digital commerce and electronic supply chain management, the value of their electronic information has increased correspondingly and the potential threats, which could compromise it, have multiplied. With the advent of networking, enterprise-critical applications, multi-tiered architectures and web access, approaches to security have become far more sophisticated. A span of research from authorization [9, 28, 14], to inference control [1], to multilevel secure databases [33, 31], and to multi-level secure transaction processing [3], addresses primarily how to protect the security of a database, especially its confidentiality. However, limited solutions has been presented on how to practically implement a solution to survive successful database attacks, which can seriously impair the integrity and availability of a database. Experience with data-intensive applications such as credit card billing, has shown that a variety of attacks do succeed to fool traditional database protection mechanisms. One critical step towards attack resistant database systems is intrusion detection, which has attracted many researchers [7, 21, 13, 10, 23, 26, 22, 17, 18]. Intrusion detection systems monitor system or network activity to discover attempts to disrupt or gain illicit access to systems. The methodology of intrusion detection can be roughly classed as being either based on statistical profiles [15, 16, 30] or on known patterns of attacks, called signatures [11, 8, 27, 12, 32]. Intrusion detection can supplement protection of network and information systems by rejecting the future access of detected attackers and by providing useful hints on how to strengthen the defense. However, intrusion detection has several inherent limitations: Intrusion detection makes the system attack-aware but not attack-resistant, that is, intrusion detection itself cannot maintain the integrity and availability of the database in face of attacks. Achieving accurate detection is usually difficult or expensive. The false alarm rate is high in many cases. The average detection latency in many cases is too long to effectively confine the damage. To overcome the limitations of intrusion detection, a broader perspective is introduced, saying that in addition to detecting attacks, countermeasures to these successful attacks should be planned and deployed in advance. In the literature, this is referred to as survivability or intrusion tolerance. In this paper, we will address a useful technique for database intrusion prevention, and present the design of a practical system, which can do attack prevention.

2 Problem Formulation

In order to protect information stored in a database, it is known to store sensitive data encrypted in the database. To access such encrypted data you have to decrypt it, which could only be done by knowing the encryption algorithm and the specific decryption key being used. The access to the decryption keys could be limited to certain users of the database system, and further, different users could be given different access rights. Specifically, it is preferred to use a so-called granular security solution for the encryption of databases, instead of building walls around servers or hard drives. In such a solution, which is described in this paper, a protective layer of encryption is provided around specific sensitive data-items or objects. This prevents outside attacks as well as infiltration from within the server itself. This also allows the security administrator to define which data stored in databases are sensitive and thereby focusing the protection only on the sensitive data, which in turn minimizes the delays or burdens on the system that may occur from other bulk encryption methods. Most preferably the encryption is made on such a basic level as in the column level of the databases. Encryption of whole files, tables or databases is not so granular, and does thus encrypt even non-sensitive data. It is further possible to assign different encryption keys of the same algorithm to different data columns. With multiple keys in place, intruders are prevented from gaining full access to any database since a different key could protect each column of encrypted data.

2.1 New Requirements

The complexity of this task was dramatically increased by the introduction of multi-platform integrated software solutions, the proliferation of remote access methods and the development of applications to support an increasing number of business processes. In the "good old days", files and databases contained fewer types of information (e.g., payroll or accounting data) stored in centralized locations, which could only be accessed, by a limited number of individuals using a handful of controlled access methods. As more types of information were migrated to electronic formats (and ever more databases proliferated, often with little planning), there was a simultaneous increase in the number of users, access methods, data flows among components and the complexity of the underlying technology infrastructure. Add to this the demand from users forever more sophisticated uses of information (data mining, CRM, etc.), which are still evolving, and the management's enhanced awareness of the value of its information. Database intrusion tolerance can mainly be enforced at two possible levels: database level and transaction level. Although transaction level methods cannot handle database level attacks, it is shown that in many applications where attacks are enforced mainly through malicious transactions transaction level methods can tolerate intrusions in a much more effective and efficient way. Database level intrusion tolerance techniques can be directly integrated into an intrusion tolerance framework with the ability to back out from a malicious database transaction. Two levels of intrusion response behavior may be deployed; an intrusion into the database system as such, or an intrusion to the actual data. In the first case focus is on preventing from further malicious activities, i e you have had an attack but it is handled by next layer of security. In the second the behavior is a rollback of the data written, to handle the attack afterwards. The importance of privacy and security of sensitive data stored in relational databases is fueled by strong new legislation and the continuing push toward Web-accessible data. These products and services allow organizations to comply with data-privacy regulations, requirements and guidelines such as the recently enacted U.S. Gramm-Leach-Bliley Act (GLBA)…significantly affecting financial institutions and insurance companies; the U.S. Health Information Portability and Accountability Act (HIPAA)…covering the healthcare industry; the European Directive 95/46/EC on data protection, and E.U./U.S. Safe Harbor considerations; Canada’s Personal Information Protection and Electronic Document Act (PIPEDA); Germany's Federal Data Protection Act; the UK Data Protection Act; Australia’s Privacy Act); the Japan JIS Q 15001:1999 Requirements for Compliance Program on Personal Information Protection; the U.S. Software and Information Industry Association (SIIA) -An Electronic Citadel - A Method for Securing Credit Card and Private Consumer Data in E-Business Sites; the BITS (the technology group for the Financial Services

Roundtable) Voluntary Guidelines for Aggregation Services; and potentially much more..

3 Problem Solution

In the above-mentioned solutions the security administrator is responsible for setting the user permissions. Thus, for a commercial database, the security administrator operates through a middle-ware application, the access control system (ACS), which provides authentication, encryption and decryption services. The ACS is tightly coupled to the database management system (DBMS) of the database. The ACS controls access in real-time to the protected elements of the database. Such a security solution provides separation of the duties of a security administrator from a database administrator (DBA). The DBA’s role could for example be to perform usual DBA tasks, such as extending tablespaces etc, without being able to see (decrypt) sensitive data. The SA could then administer privileges and permissions, for instance add or delete users. For most commercial databases, the database administrator has privileges to access the database and perform most functions, such as changing password of the database users, independent of the settings by the system administrator. An administrator with root privileges could also have full access to the database. This is an opening for an attack where the DBA can steal all the protected data without any knowledge of the protection system above. The attack is in this case based on that the DBA impersonates another user by manipulating that users password, even though a hash algorithm enciphers the user’s password. An attack could proceed as follows. First the DBA logs in as himself, and then the DBA reads the hash value of the users password and stores this separately. Preferably the DBA also copies all other relevant user data. By these actions the DBA has created a snapshot of the user before any altering. Then the DBA executes the command “ALTER USER username IDENTIFIED BY newpassword”. The next step is to log in under the user name "username” with the password “newpassword” in a new session. The DBA then resets the user’s password and other relevant user data with the previously stored hash value. Thus, it is important to further separate the DBA’s and the SA’s privileges. For instance, if services are outsourced, the owner of the database contents may trust a vendor to administer the database. Then the role of the DBA belongs to an external person, while the important SA role is kept within the company, often at a high management level. Thus, there is a need for preventing a DBA to impersonate a user in an attempt to gain access to the contents of the database. The DBA attack prevention described here is specific to databases with internal authentication. Databases that utilizes external (OS level) authentication provides a level of separation of duties, and the database encryption system, or intrusion prevention system, can verify that the database session is properly authenticated by the external authentication system before any decryption of sensitive data is allowed.

3.1 A New Approach

The solution protects the data in storage in a database. The architecture is built on top of a traditional COTS (Commercial-Of-The-Shelf) DBMS. Within the framework, the Intrusion Detector identifies malicious transactions based on the history kept (mainly) in the log. The Intrusion Assessor locates the damage caused by the detected transactions. The Intrusion Protector prevents the damage using a rollback. The Intrusion Manager restricts the access to the objects that have been identified by the Intrusion Assessor as ‘under attack’, and unlocks an object after it is cleared by the security officer. The Policy Enforcement Agent (PEA) (a) functions as a filter for normal user transactions that access critical fields in the database, and (b) is responsible for enforcing system-wide intrusion prevention policies. For example, a policy may require the PEA to reject every new transaction submitted by a user as soon as the Intrusion Detector finds that the user submits a malicious transaction. It should be noticed that the system is designed to do all the intrusion prevention work on the fly without the need to periodically halt normal transaction processing.

3.2 Intrusion Prevention Solution

The method allows for a real time prevention of intrusion by letting the intrusion detection process interact directly with the access control system, and change the user authority dynamically as a result of the detected intrusion. The hybrid solution combines benefits from database encryption toolkits and secure key management systems. The hybrid solution also provides a single point of control for database intrusion prevention, audit, privacy policy management, and secure and automated encryption key management (FIPS 140 Level 3). The Database Intrusion Prevention is based on ‘context checking’ against a protection policy for each critical database column, and prevents internal attacks also from root, DBA, or ‘buffer overflow attacks’, by automatically stopping database operations that are not conforming to the Database Intrusion Prevention Policy rules. The Database Intrusion Prevention and alarm system enforces policy rules that will keep any malicious application code in a sand box regarding database access. The policy enforcement system, integrated with an external network authentication system, perform the following basic checking:

Session Authentication and Session Encryption.
Software Integrity, Data Integrity, and Meta Data Integrity.
Time of Access, and other policy rules.

In database security, it is a well-known problem to avoid attacks from persons who have access to a valid user-ID and password. Such persons cannot be denied access by the normal access control system, as they are in fact entitled to access to a certain extent. Such persons can be tempted to access improper amounts of data, by-passing the security. Such persons can be monitored and controlled by this database intrusion prevention system and automatically be locked out from database operations that are not conforming to the Database Intrusion Prevention Policy rules. Other solutions in this problem area have been suggested:

Network-Based Detection - Network intrusion monitors are attached to a packet-filtering router or packet sniffer to detect suspicious behavior on a network as they occur. They look for signs that a network is being investigated for attack with a port scanner, that users are falling victim to known traps like .url or .lnk, or that the network is actually under an attack such as through SYN flooding or unauthorized attempts to gain root access (among other types of attacks). Based on user specifications, these monitors can then record the session and alert the administrator or, in some cases, reset the connection. Some examples of such tools include Cisco's NetRanger and ISS' RealSecure as well as some public domain products like Klaxon that focus on a narrower set of attacks.

Server-Based Detection - These tools analyze log, configuration and data files from individual servers as attacks occur, typically by placing some type of agent on the server and having the agent report to a central console. Some examples of these tools include Axent's OmniGuard Intrusion Detection (ITA), Security Dynamic's Kane Security Monitor and Centrax's eNTrax as well as some public domain tools that perform a much narrower set of functions like Tripwire which checks data integrity. Tripwire will detect any modifications made to operating systems or user files and send alerts to ISS' RealSecure product. Real-Secure will then conduct another set of security checks to monitor and combat any intrusions.

Security Query and Reporting Tools - These tools query NOS logs and other related logs for security events or they glean logs for security trend data. Accordingly, they do not operate in real-time and rely on users asking the right questions of the right systems. A typical query might be how many failed authentication attempts have we had on these NT servers in the past two weeks." A few of them (e.g., SecurIT) perform firewall log analysis. Some examples of such tools include Bindview's EMS/NOSadmin and Enterprise Console, SecureIT's SecureVIEW and Security Dynamic's Kane Security Analyst.

3.3 Inference Detection

A variation of conventional intrusion detection is detection of specific patterns of information access, deemed to signify that an intrusion is taking place, even though the user is authorized to access the information. A method for such inference detection, i.e. a pattern oriented intrusion detection, is disclosed in US patent 5278901 to Shieh et al.
None of these solutions are however entirely satisfactory. The primary drawback is that they all concentrate on already effected queries, providing at best information that an attack has occurred.

3.4 Intrusion Prevention Profile

By defining at least one intrusion detection profile, each comprising at least one item (column access) access rate, associating each user with one of the profiles, receiving a query from a user, comparing a result of the query with the item access rates defined in the profile associated with the user, determining whether the query result exceeds the item access rates, and in that case notifying the access control system to alter the user authorization, thereby making the received request an unauthorized request, before the result is transmitted to the user. According to this method, the result of a query is evaluated before it is transmitted to the user. This allows for a real time prevention of intrusion, where the attack is stopped even before it is completed. This is possible by letting the intrusion detection process interact directly with the access control system, and change the user authority dynamically as a result of the detected intrusion. The item access rates can be defined based the number of rows a user may access from an item, e.g. a column in a database table, at one time, or over a certain period of time. In a preferred implementation, the method further comprises accumulating results from performed queries in a record, and determining whether the accumulated results exceed any one of the item access rates. The effect is that on one hand, a single query exceeding the allowed limit can be prevented, but so can a number of smaller queries, each one on its on being allowed, but when accumulated not being allowed. It should be noted that the accepted item access rates not necessarily are restricted to only one user. On the contrary, it is possible to associate an item access rate to a group of users, such as users belonging to the same access role (which defines the user's level of security), or connected to the same server. The result will be restricting the queries accepted from a group of users at one time or over a period of time. The user, role and server entities are not exclusive of other entities which might benefit from a security policy. According to an implementation of the method, items subject to item access rates are marked in the database, so that any query concerning the items automatically can trigger the intrusion detection process. This is especially advantageous if only a few items are intrusion sensitive, in which case most queries are not directed to such items. The selective activation of the intrusion detection will then save time and processor power. According to another implementation of the method, the intrusion detection policy further includes at least one inference pattern, and results from performed queries are accumulated in a record, which is compared to the inference pattern, in order to determine whether a combination of accesses in the record match the inference policy, and in that case the access control system is notified to alter the user authorization, thereby making the received request an unauthorized request, before the result is transmitted to the user. This implementation provides a second type of intrusion detection, based on inference patterns, again resulting in a real time prevention of intrusion.

4 Related Work

There is a variety of related research efforts that explore what one can do with audit data to automatically detect threats to the host. An important work is MIDAS [50], as it was one of the original applications of expert systems—in fact using P-BEST—to the problem of monitoring user activity logs for misuse and anomalous user activity. CMDS, by SAIC, demonstrated another application of a forward-chaining expert-system, CLIPS, to a variety of operating system logs [48]. USTAT [39] offered another formulation of intrusion heuristics using state transition diagrams [46], but by design remained a classic forward-chaining expert sys-tem inference engine. ASAX [37] introduced the Rule-based Sequence Evaluation Language (RUSSEL) [42], which is tuned specifically for the analysis of host audit trails. Recent literature form the RAID conferences, as well as IEEE Security and Privacy, the DARPA program on survivability that concentrated on detecting and surviving attacks, and a large scale DARPA project called DemVal, are dealing with the survivability of a database. The idea of attack prevention, that will not allow access after a threshold is reached, is also discussed in the SRI Appache IDs system. The approach is sometimes also called application level intrusion detection, rather than procedural intrusion detection.

5 Conclusion

While the existing paradigms of computer security are still very useful and serve perfectly well in their capacities, there has existed a gap in the computer security space. Our technology and approach fills that gap by providing practical application based intrusion detection and response. We suggest that this gives The Hybrid the unique ability to detect and halt completely novel attacks that have yet to be seen on the Internet, and better yet, we have the ability to protect the first person to see a new attack or exploit. No one needs to be sacrificed to the new virus or worm anymore. In essence, we have learned to solve the right problem. Removing all software vulnerabilities is clearly an unsolvable problem. Providing restrictive and onerous barriers to software use makes the software uncomfortable and difficult to use. Monitoring and controlling program execution at run time through behavioral control is the missing piece in the security puzzle. The complete puzzle has three pieces; data control (encryption), access control, and behavioral control.

In conclusion, while the overall complexity of the security program has dramatically increased, enterprises can still implement effective security solutions by integrating sound external protection and internal security controls with appropriate security audit procedures. There are no guarantees that any one approach will be able to deal with new and innovative intrusions in increasingly complex technical and business environments. However, implementation of an integrated security program which is continuously audited and monitored provides the multiple layers of protection needed to maximize protection as well as historical information to support management decision-making and future policy decisions. This solution protects the data during transport, providing security from the server to the client. The client device requires a means of accessing the secure data, and a means of access control and secure storage of locally held information. The implementation for Laptops and PDAs provides mandatory access control, secure local storage of sensitive data and key management capabilities. This solution includes a method for detecting intrusion in a database, managed by an access control system, comprising defining at least one intrusion detection profile, each comprising at least one item access rate and associating each user with one of the profiles. Further, the method determines whether a result of a query exceeds any one of the item access rates defined in the profile associated with the user, and, in that case, notifies the access control system to alter the user authorization, thereby making the received request an unauthorized request, before the result is transmitted to the user. The method allows for a real time prevention of intrusion by letting the intrusion detection process interact directly with the access control system, and change the user authority dynamically as a result of the detected intrusion.

The GLBA/OCC and the VISA U.S.A. CISP requirements as well as other requirements in the Health Care Industry, and Safe Harbor will require a unique demonstration of cooperative and open but protected communication, storing information among individuals and organizations across competitive lines and regulatory boundaries safeguarding non-public personal information. Information sharing among reliable and reputable experts can help institutions reduce the risk of information system intrusions. The OCC encourages management to participate in information-sharing mechanisms as part of an effort to detect and respond to intrusion and vulnerabilities. Financial institutions have to work together in an unprecedented fashion with other financial institutions, service providers, software vendors, trade associations, regulators, and other industries to share information and strategies to respond to legal requirements and media reports or perceptions that could decrease public confidence in the financial services industry. With the introduction of regulatory privacy acts like the U.S. Gramm-Leach-Bliley Act, the U.S. HIPAA, the U.S. FDA 21 CFR 11 and the E.U. member states privacy laws, companies are being mandated to provide more detailed information regarding the usage and access of customer and consumer data.

References

[1] M. R. Adam. *Security-Control Methods for Statistical Database: A Comparative Study.* ACM Computing Surveys, 21(4), 1989.

[2] P. Ammann, S. Jajodia, and P. Liu. *Recovery from malicious trans-actions.*

IEEE Transactions on Knowledge and Data Engineering, 2001. To appear.

[3] V. Atluri, S. Jajodia, and B. George. *Multilevel Secure Transaction Processing.* Kluwer Academic Publishers, 1999.

[4] D. Barbara, R. Goel, and S. Jajodia. *Using checksums to detect data corruption.* In Proceedings of the 2000 International Conference on Extending Data Base Technology, Mar 2000.

[5] P. A. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems.* Addison-Wesley, Reading, MA, 1987.

[6] S. B. Davidson. *Optimism and consistency in partitioned distributed database systems*. ACM Transactions on Database Systems, 9(3):456–581, September 1984.

[7] D.E.Denning. *An intrusion-detection model.* IEEE Trans. on Software Engineering, SE-13:222–232, February 1987.

[8] T.D. Garvey and T.F. Lunt. *Model-based intrusion detection*. In Proceedings of the 14th National Computer Security Conference, Balti-more, MD, October 1991.

[9] P. P. Griffiths and B. W. Wade. *An Authorization Mechanism for a Relational Databas*e System. ACM Transactions on Database Systems, 1(3):242–255, September 1976.

[10] P. Helman and G. Liepins. *Statistical foundations of audit trail analysis for the detection of computer misuse.* IEEE Transactions on Software Engineering, 19(9):886–901, 1993.

[11] K. Ilgun. Ustat: *A real-time intrusion detection system for unix.* In Proceedings of the IEEE Symposium on Security and Privacy,Oak-land, CA, May 1993.

[12] K. Ilgun, R.A. Kemmerer, and P.A. Porras. *State transition analysis*: A rule-based intrusion detection approach. IEEE Transactions on Software Engineering, 21(3):181–199, 1995.

[13] R. Jagannathan and T. Lunt. *System design document: Next generation intrusion detection expert system* (nides). Technical report, SRI International, Menlo Park, California, 1993.

[14] S. Jajodia, P. Samarati, V. S. Subrahmanian, and E. Bertino. *A unified framework for enforcing* multiple access control policies. In Proceedings of ACM SIGMOD International Conference on Management of Data, pages 474–485, May 1997.

[15] H. S. Javitz and A. Valdes. The sri ides statistical anomaly detector. In Proceedings IEEE Computer Society Symposium on Security and Privacy, Oakland, CA, May 1991.

[16] H. S. Javitz and A. Valdes. The nides statistical component description and justification. Technical Report A010, SRI International, March 1994.

[17] T. Lane and C.E. Brodley. Temporal sequence learning and data reduction for anomaly detection. In Proc. 5th ACM Conference on Computer and Communications Security, San Francisco, CA, Nov 1998.

[18] Wenke Lee, Sal Stolfo, and Kui Mok. A data mining framework for building intrusion detection models. In Proc. 1999 IEEE Symposium on Security and Privacy, Oakland, CA, May 1999.

[19] P. Liu, S. Jajodia, and C.D. McCollum. Intrusion confinement by isolation in information systems. Journal of Computer Security, 8(4):243–279, 2000.

[20] P. Luenam and P. Liu. Odam: An on-the-fly damage assessment and repair system for commercial database applications. In Proc. 15th IFIP WFG11.3 Working Conference on Database and Application Security, Ontario, Canada, July 2001.

[21] T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalali, H. S. Javitz, A. Valdes, P. G. Neumann, and T. D. Garvey. A real time intrusion detection expert system (ides). Technical report, SRI International, Menlo Park, California, 1992.

[22] Teresa Lunt and Catherine McCollum. Intrusion detection and response research at DARPA. Technical report, The MITRE Corporation, McLean, VA, 1998.

[23] T.F. Lunt. A Survey of Intrusion Detection Techniques. Computers & Security, 12(4):405–418, June 1993.

[24] J. McDermott and D. Goldschlag. Storage jamming. In D.L. Spooner, S.A. Demurjian, and J.E. Dobson, editors, Database Se-curity IX: Status and Prospects, pages 365–381. Chapman & Hall, London, 1996.

[25] J. McDermott and D. Goldschlag. *Towards a model of storage jamming.* In Proceedings of the IEEE Computer Security Foundations

Workshop, pages 176–185, Kenmare, Ireland, June 1996.

[26] B. Mukherjee, L. T. Heberlein, and K.N. Levitt. *Network intrusion detection.* IEEE Network, pages 26–41, June 1994.

[27] P.A. Porras and R.A. Kemmerer. *Penetration state transition analysis: A rule-based intrusion detection approach.* In Proceedings of the 8th Annual Computer Security Applications Conference, San Antonio, Texas, December 1992.

[28] F. Rabitti, E. Bertino, W. Kim, and D. Woelk. *A model of authorization for next generation database systems.* ACM Transactions on Database Systems, 16(1):88–131, 1994.

[29] P. Liu S. Ingsriswang. Aaid: *An application aware transaction level database intrusion detection* system. Technical report, Department of Information Systems, UMBC, Baltimore, MD, 2001.

[30] D. Samfat and R. Molva. Idamn: *An intrusion detection architecture for mobile networks.* IEEE Journal of Selected Areas in Communications, 15(7):1373–1380, 1997.

[31] R. Sandhu and F. Chen. *The multilevel relational (mlr) data model.* ACM Transactions on Information and Systems Security, 1(1), 1998.

[32] S.-P. Shieh and V.D. Gligor. On a pattern-oriented model for intrusion detection. IEEE Transactions on Knowledge and Data Engi-neering, 9(4):661–667, 1997.

[33] M. Winslett, K. Smith, and X. Qian. Formal query languages for secure relational databases. ACM Transactions on Database Systems, 19(4):626–662, 1994.

[34] The U.S. Health Information Portability and Accountability Act (HIPAA) - compliance by October 2002 www.hipaacomply.com

[35] The European Union 95/46/EC Directive on Data Privacy - compliance October 1998 - and individual EU member state privacy legislation - various compliance dates http://europa.eu.int/comm/internal_market/en/dataprot/

[36] EU/US Safe Harbor - compliance 11/1/2000 www.export.gov/safeharbor
http://europa.eu.int/comm/internal_market/en/dataprot/modelcontracts/index.htm

[37] J. Habra, B. Le Charlier, A. Mounji, and I. Mathieu. ASAX: Software architecture and rule-based language for universal audit trail analysis. In Y. Deswarte et al., editors, Computer Security – Proceedings of ESORICS 92, volume 648 of LNCS, pages 435–450, Toulouse, France, Nov. 23–25, 1992. Springer-Verlag.

[38] L. T. Heberlein et al. A network security monitor. In Proceedings of the 1990 IEEE Symposium on Security and Pri-vacy, pages 296–304, Oakland, California, May 7–9, 1990.

[39] K. Ilgun. USTAT: A real-time intrusion detection system for UNIX. In Proceedings of the 1993 IEEE Symposium on Security and Privacy, pages 16–28, Oakland, California, May 24–26, 1993.

[40] U. Lindqvist and P. A. Porras. Detecting computer and network misuse through the production-based expert system toolset (P-BEST). In Proceedings of the 1999 IEEE Symposium on Security and Privacy, pages 146–161, Oakland, California, May 9–12, 1999.

[41] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das. Analysis and results of the 1999 DARPA off-line intrusion detection evaluation. In H. Debar, L. M´ e, and S. F. Wu, editors, Recent Advances in Intrusion Detection (RAID 2000), volume 1907 of LNCS, pages 162–182, Toulouse, France, Oct. 2–4, 2000. Springer-Verlag.

[42] A. Mounji. Languages and Tools for Rule-Based Distributed Intrusion Detection. PhD thesis, Institut d'Informatique, University of Namur, Belgium, Sept. 1997.

[43] P. G. Neumann and P. A. Porras. Experience with EMERALD to date. In Proceedings of the 1st Workshop on Intrusion Detection and Network Monitoring, Santa Clara, California, Apr. 9–12, 1999. The USENIX Association.

[44] A. One. Smashing the stack for fun and profit. Phrack Magazine, 7(49), Nov. 8, 1996. http://www.fc.net/phrack/files/ p49/p49-14.

[45] J. Picciotto. The design of an effective auditing subsystem. In Proceedings of the 1987 IEEE Symposium on Security and Privacy, pages 13–22, Oakland, California, Apr. 27–29, 1987.

[46] P. A. Porras and R. A. Kemmerer. Penetration state transitionanalysis: A rule-based intrusion detection approach. In Proceedings of the Eighth Annual Computer Security Applications Conference, pages 220–229, San Antonio, Texas, Nov. 30–Dec. 4, 1992.

[47] P. A. Porras and P. G. Neumann. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In Proceedings of the 20th National Information Systems Security Conference, pages 353–365, Baltimore, Maryland, Oct. 7–10, 1997. National Institute of Standards and Tech-nology/National Computer Security Center.

[48] P. Proctor. Audit reduction and misuse detection in heterogeneous environments: Framework and application. In Proceedings of the Tenth Annual Computer Security Applications Conference, pages 117–125, Orlando, Florida, Dec. 5–9, 1994.

[49] T. H. Ptacek and T. N. Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical report, Secure Networks, Inc., Calgary, Alberta, Canada, Jan. 1998. http://www.clark.net/˜roesch/idspaper.html.

[50] M. M. Sebring, E. Shellhouse, M. E. Hanna, and R. A. Whitehurst. Expert systems in intrusion detection: A case study. In Proceedings of the 11th National Computer Security Conference, pages 74–81, Baltimore, Maryland, Oct. 17–20, 1988. National Institute of Standards and Technology/National Computer Security Center.

[51] Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, CA 94303, USA. SunSHIELD Basic Security Module Guide, Solaris 7, Oct. 1998. Part No. 805-2635-10.

[52] U.S. Department of Defense. Trusted Computer System Evaluation Criteria, Dec. 1985. DoD 5200.28-STD.

[53] A. Valdes and K. Skinner. Adaptive, model-based monitoring for cyber attack detection. In H. Debar, L. M´ e,and S. F. Wu, editors, Recent Advances in Intrusion De-tection (RAID

2000), volume 1907 of LNCS, pages 80–92,Toulouse, France, Oct. 2–4, 2000. Springer-Verlag.

[54] U. T. Mattsson, and T. Valfridsson. An automated method to minimize the risk for exposure of encryption keys and encrypted database information. EPC Patent number – 00/975134.8.

[55] U. T. Mattsson. A method for implementation of encryption in a 24 by 7 production database. US Patent number 09/712 926.

[56] U. T. Mattsson. A method for detecting and preventing intrusions in commercial databases. EPC Patent number EP 01127906.4.

[57] U. T. Mattsson. A method for protecting databases against internal attacks. Sweden Patent number 0004189-7.

[58] U. T. Mattsson. Basic Data Type transparent method for storing and transporting of encrypted data. US Patent number 09/721 942.

[59] U. T. Mattsson. A method for combining software based encryption and hardware based encryption and key management. US Patent number 09/712 941.

[60] UK's Data Protection Act - Compliance March 1, 2000 www.dataprotection.gov.uk

[61] Canada’s Personal Information Protection and Electronic Document Act (PIPEDA) Compliance 1/1/2001 to 1/1/2004 www.privcom.gc.ca

[42] Australia’s Privacy Act – Compliance by December 21, 2001 www.privacy.gov.au

[63] The VISA U.S.A. Cardholder Information Security Program (CISP) – Compliance May 1, 2001http://usa.visa.com/business/merchants/cisp_indexhtml

[64] The VISA International Account Information Security Standards (AIS) and Best Practices Guide https://www.visa.com/nt/gds/main.html

[65] The U.S. Software and Information Industry Association (SIIA) - An Electronic Citadel - A Method for Securing Credit Card and Private Consumer Data in E-Business Sites www.siia.net/sharedcontent/divisions/ebus/citadel.pdf

[66] The BITS (the technology group for the Financial Services Roundtable) Voluntary Guidelines for Aggregation Services www.bitsinfo.org/FinalAggregationBook051601.pdf

[67] The U.S. Gramm-Leach-Bliley Act (GLBA) (TITLE V--Consumer Privacy), regulated by the SEC, FTC, FDIC, OCC, OTS, FRB, NAIC, and NCUA, which covers a broad range of financial services and virtually affects any company who accepts credit cards - compliance July 1st, 2001 www.complianceheadquarters.com/Privacy/Privacy_Research/privacy_research.html

[68] EU member state privacy legislations see http://europa.eu.int/comm/internal_market/en/dataprot/law/impl.htm

[69] Germany's Federal Data Protection Act (Der Bundesbeauftragte für den Datenschutz) - compliance May 23, 2001 www.bfd.bund.de

[70] Sweden's Personal Data Act (Personuppgiftslagen - PuL) - compliance October 1, 2001 www.datainspektionen.se

Efficient Tree Search in Encrypted Data

R. Brinkman, L. Feng, J. Doumen, P.H. Hartel, and W. Jonker

University of Twente, Enschede, the Netherlands
{brinkman,ling,doumen,pieter,jonker}@cs.utwente.nl

Abstract. Sometimes there is a need to store sensitive data on an untrusted database server. Song, Wagner and Perrig have introduced a way to search for the existence of a word in an encrypted textual document. The search speed is linear in the size of the document. It does not scale well for a large database. We have developed a tree search algorithm based on the linear search algorithm that is suitable for XML databases. It is more efficient since it exploits the structure of XML. We have built prototype implementations for both the linear and the tree search case. Experiments show a major improvement in search time.

1 Introduction

Nowadays the need grows to store data securely on an untrusted system. Think, for instance, of a remote database server administered by somebody else. If you want your data to be secret, you have to encrypt it. The problem then arises how to query the database. The most obvious solution is to download the whole database locally and then perform the query. This of course is terribly inefficient. Song, Wagner and Perrig [1] have introduced a protocol to search for a word in an encrypted text. We will summarise this protocol in section 2.

In this paper we propose a new protocol that is more suitable for handling large amounts of semi-structured XML data. This new protocol exploits the XML tree structure. XPath queries can be answered fast and secure.

We have built prototype implementations for both the linear and the tree search protocol (section 3). We use these prototypes to find optimal settings for the parameters used within the protocols and to show the increase in search speed by using the tree structure. We did some experiments (section 4) for which the results can be found in section 5.

2 Search Strategy

Before we describe our tree search strategy (section 2.2) we will give a short summary of the original linear search strategy of Song, Wagner and Perrig [1].

2.1 Linear Search Strategy for Full Text Documents

Song, Wagner and Perrig [1] describe a protocol to store sensitive data on an untrusted server. A client (Alice) can store data on the untrusted server (Bob) and search in it, without revealing the plain text of either the stored data, the query or the query result. The protocol consists of three parts: storage, search and retrieval.

Storage Before Alice can store information on Bob she has to do some calculations. First of all she has to fragment the whole plain text W into several fixed sized words W_i. Each W_i has a fixed length n. She also generates encryption keys k' and k'' and a sequence of random numbers S_i using a pseudo random generator. Then she has or calculates the following for each block W_i:

W_i	plain text block
k''	encryption key
$X_i = E_{k''}(W_i) = \langle L_i, R_i \rangle$	encrypted text block
k'	key for f
$k_i = f_{k'}(L_i)$	key for F
S_i	random number i
$T_i = \langle S_i, F_{k_i}(S_i) \rangle$	tuple used by search
$C_i = X_i \oplus T_i$	value to be stored ($\oplus$ stands for xor)

where E is an encryption function and f and F are keyed hash functions:

$$\begin{aligned} E &: \mathit{key} \times \{0,1\}^n \rightarrow \{0,1\}^n \\ f &: \mathit{key} \times \{0,1\}^{n-m} \rightarrow \mathit{key} \\ F &: \mathit{key} \times \{0,1\}^{n-m} \rightarrow \{0,1\}^m \end{aligned}$$

The encrypted word X_i has the same block length as W_i (i.e. n). L_i has length $n - m$ and R_i has length m. The parameters n and m may be chosen freely ($n > 0$, $0 < m \leq \frac{n}{2}$). Section 5.1 gives guidelines for efficient values of n and m. The value C_i can be sent to Bob for storage. Alice may now forget the values W_i, X_i, L_i, R_i, k_i, T_i and C_i, but should still remember k', k'' and S_i.

Search After the encrypted data is stored by Bob in the previous phase Alice can query Bob. Alice provides Bob with an encrypted version of a plain text word W_j and asks him if and where W_j occurs in the original document. Note that Alice does not have to know the position j. If W_j was a block in the original data then $\langle j, C_j \rangle$ is returned. Alice has or calculates:

k''	encryption key
k'	key for f
W_j	plain text block to search for
$X_j = E_{k''}(W_j) = \langle L_j, R_j \rangle$	encrypted block
$k_j = f_{k'}(L_j)$	key for F

Then Alice sends the value of X_j and k_j to Bob. Having X_j and k_j Bob is able to compute for each C_p:

$$\begin{aligned} & T_p = C_p \oplus X_j = \langle S_p, S'_p \rangle \\ & \text{IF } S'_p = F_{k_j}(S_p) \text{ THEN RETURN } \langle p, C_p \rangle \end{aligned}$$

If $p = j$ then $S'_p = F_{k_j}(S_p)$, otherwise S'_p is garbage. Note that all locations with a correct T_p value are returned. However there is a small chance that T satisfies $T = \langle S_q, F_{k_j}(S_q)\rangle$ but where $S_q \neq S_p$. Therefore, Alice should check each answer whether the correct random value is used or not.

Retrieval Alice can also ask Bob for the cipher text C_p at any position p. Alice, knowing k', k'' and the seed for S, can recalculate W_p by

p	desired location
$C_p = \langle C_{p,l}, C_{p,r}\rangle$	stored block
S_p	random value
$X_{p,l} = C_{p,l} \oplus S_p$	left part of encrypted block
$k_p = f_{k'}(X_{p,l})$	key for F
$T_p = \langle S_p, F_{k_p}(S_p)\rangle$	check tuple
$X_p = C_p \oplus T_p$	encrypted block
$W_p = D_{k''}(X_p)$	plain text block

where D is the decryption function $D : \mathit{key} \times \{0,1\}^n \rightarrow \{0,1\}^n$ such that $D_{k''}(E_{k''}(W_i)) = W_i$.

This is all Alice needs. She can store, find and read the text while Bob cannot read anything of the plain text. The only information Bob gets from Alice is C_i in the store phase and X_j and k_j in the search phase. Since C_i and X_j are both encrypted with a key only known to Alice and k_j is only used to hash one particular random value, Bob does not learn anything of the plain text. The only information Bob learns from a search query is the location where an encrypted word is stored.

2.2 Tree Search Strategy for XML Documents

So far, we considered only text files. Using structured XML data can improve efficiency.

Torsten Grust [2, 3] introduces a way to store XML data in a relational database such that search queries can be handled efficiently. An XML document is translated into a relational table with a predefined structure. Each record consists of the name of the tag or attribute and its corresponding value. The information about the tree structure of the original XML document is captured in the pre, post and parent fields. All fields can be computed in a single pass over the XML document. The pre and post fields are sequence numbers that count the open tags respectively the close tags. The parent value is the pre value of the parent element (see figure 1(a)).

The XPath axes like *descendant*, *ascendant*, *child*, etc can be expressed as simple expressions over the pre, post and parent fields. For instance:

- v is a child of $v' \iff v.\mathit{parent} = v'.\mathit{pre}$
- v is a descendant of $v' \iff v'.\mathit{pre} < v.\mathit{pre} \wedge v'.\mathit{post} > v.\mathit{post}$
- v is following $v' \iff v'.\mathit{pre} < v.\mathit{pre} \wedge v'.\mathit{post} < v.\mathit{post}$

	pre	post	parent
<a>	1		0
<b>	2		1
</b>		1	
<c	3		1
d=”. . . ”>	4	2	3
<e/>	5	3	3
</c>		4	
</a>		5	

(a) Pre/Post/Parent calculation

(b) Visualisation of XPath Axes in a Pre/Post Plane

Fig. 1. Calculation and Usage of Pre, Post and Parent fields

Some XPath axes can also be drawn in a pre/post plane (see figure 1(b)). Each element can be drawn as a dot in the graph. The solid circle indicates just one of them. Taking the solid circle as starting element, the quadrants indicate where its ascendants, descendants and siblings are located.

Not all updates are efficient. Modification and deletion are no problem, but element insertion causes the need to recalculate the pre, post and parent values for all following elements. The number of recalculations can be reduced by an initial sequence with a larger step (100, 200, 300, . . .).

Torsten Grust aims at storing XML data in the clear. To protect the data cryptographically we combine his strategy with the linear search approach of Song, Wagner and Perrig (SWP) [1]. Only some slight modifications to the SWP approach are necessary:

1. The input file is not an unstructured text file but a tree structured XML document. The division of the data into fixed sized blocks does not seem natural. Therefore, we use variable block lengths that depend on the lengths of the tag names, attribute names, attribute values and the text between tags.
2. The sequence number of a block is no longer appropriate to define the location within a document. We use the pre value instead.

The equations of section 2.1 can be rewritten to the equations below. Note that all subscripts have changed. For simplicity we only describe the encryption of tag names. Exactly the same scheme is used for attribute names (prefixed with a @ sign) or the data itself by simply substituting value for tag.

Storage

W_{tag}	plain text block
k''	encryption key
$X_{tag} = E_{k''}(W_{tag}) = \langle L_{tag}, R_{tag} \rangle$	encrypted text block
k'	key for f
$k_{tag} = f_{k'}(L_{tag})$	key for F
S_{pre}	random number pre
$T_{pre,tag} = \langle S_{pre}, F_{k_{tag}}(S_{pre}) \rangle$	tuple used by search
$C_{pre,tag} = X_{tag} \oplus T_{pre,tag}$	value to be stored

Note that the random value S_{pre} does not depend on the tag name but on the location (expressed in the pre field) because all elements with the same tag name should be stored differently.

Search An XPath query like $/tag_1//tag_2[tag_3 = "value"]$ is encrypted to $/\langle X_{tag_1}, k_{tag_1} \rangle // \langle X_{tag_2}, k_{tag_2} \rangle [\langle X_{tag_3}, k_{tag_3} \rangle = "\langle X_{value}, k_{value} \rangle"]$ before sending it to the server. The server calculates the result traversing the XPath query from left to right. Each step consists of two or three sub steps:

- Evaluating the XPath axis /, //, [and] using the pre, post and parent fields. It is possible to find all children (/) or all descendants (//) of elements found in a previous step by just using the pre, post and parent field. See section 3.2 for an example.
- Filtering out the records that do not satisfy $S'_p = F_{k_{tag}}(S_p)$ in $T_{p,tag} = C_{p,tag} \oplus X_{tag} = \langle S_p, S'_p \rangle$.
- Eventually filtering out the records with an incorrect value field.

Retrieval

k'	key for f
k''	encryption key
pre	desired location
$C_{pre,tag} = \langle C_{pre,tag,l}, C_{pre,tag,r} \rangle$	stored block
S_{pre}	random value
$X_{tag,l} = C_{pre,tag,l} \oplus S_{pre}$	left part of encrypted block
$k_{tag} = f_{k'}(X_{tag,l})$	key for F
$T_{tag} = \langle S_{pre}, F_{k_{tag}}(S_{pre}) \rangle$	check tuple
$X_{tag} = C_{pre,tag} \oplus T_{tag}$	encrypted block
$W_{tag} = D_{k''}(X_{tag})$	plain text block

3 Implementation

For each search strategy a prototype has been developed. Each prototype consists of two tools: one for encryption and one for searching. All tools use the standard crypto packages shipped with JDK 1.4.

3.1 Linear Search Prototype

Section 2.1 introduces three functions: E, f and F. E should be a block cipher in ECB mode and f and F keyed hash functions. For our prototype we chose

DES for all three of them. E is exactly DES in ECB mode. Since DES works on blocks of 64 bits n should be a multiple of 64 bits.

f and F are keyed hash functions with variable sized hash values. Standard hash functions like SHA-1 have a fixed sized hash value. It is possible to use the last (or the first) m bits of the hash value, but then m should be less than the size of the hash value (160 bits for SHA-1). To allow a larger value for m our prototype uses DES in CBC mode. To hash a data block of length $n - m$ to a hash value of length m the block is encrypted with the specified key (56 bits DES key) but only the last m bits are used as hash value. The only restriction for m is that $n - m \geq m$ and thus $n \geq 2m$. See Menezes et al [4] for a more detailed description of the used hash algorithm.

The search algorithm implements the protocol described in [1] as summarised in section 2.1. The program takes the whole cipher text along with the query as input and produces the $\langle i, C_i \rangle$ pairs as output.

3.2 Tree Search Prototype

Like the linear prototype the tree search prototype is split into two parts: one for encryption and one for searching.

The Encrypt tool uses a SAX parser to read the input XML document. In one pass over the input, the pre, post and parent values can be calculated. When an end tag is encountered all the information to encrypt the element is available. Attributes are handled as tags with a leading @ sign. A new record $\langle pre, post, parent, C_{pre,tag}, C_{pre,value} \rangle$ is inserted into the relational database, where $C_{pre,tag}$ and $C_{pre,value}$ are calculated as in section 2.2. In our prototype we use a MySQL database to store the encrypted document.

In contrast with the linear prototype there are no predefined block sizes n and m. Instead of using a fixed sized block, n is simply set to the length of the tag name. m is a predefined fraction of n (for example 0.5).

In order to speed up the search process, indices are added to the MySQL table for the pre, post and parent fields.

The XPath expression is evaluated step by step. Preliminary results are stored in a result table. Each step consists of two or three sub steps:

1. Carry out the path delimiter (/, //, [or]). For this step only the pre, post and parent fields are needed. For example // (descendants) is translated into the SQL query:

```
CREATE TABLE new_result
SELECT data.*
FROM data, previous_result
WHERE data.pre  > previous_result.pre AND
      data.post < previous_result.post
```

2. Filter out the records in the preliminary result with the wrong tag/attribute names. In this step we use the original linear search method.

3. When the step consists of an equation expression the previous step is repeated but now for the value instead of the name.

4 Experimental Data

The two prototypes give us the opportunity to experiment with the parameters used in the protocol and, more importantly, compare the linear search approach with the tree search approach. We are especially interested in the influence the approach and the parameters n and m have on the encryption and search speed. We used the XML benchmark[1] [5] to generate three sample XML files of sizes 1 MB, 10 MB and 100 MB. Although the linear approach does not use the structure of these XML files the benchmark is used in both cases to compare the results with the tree search approach.

Also the number of collisions has been measured (see figure 2(a)). Collisions are the false hits that occur because of the collisions in the hash function F. F hashes the random value S_i of size $n-m$ to a hash value of length m, where $n-m \geq m$. Therefore collisions are unavoidable (collisions are avoidable when $n-m = m$ and F is bijective, but bijective functions are not good hash functions).

4.1 Experiments with the Linear Search Prototype

For the linear prototype both n and m may be chosen freely. Tests are carried out $\forall n \in \{8, 16, 24, 32, 40, 48, 56, 64\}$ where these values are the number of bytes and not bits. Because we use DES in ECB mode for the encryption function E, we only use multiples of 8 bytes. m should be less than or equal to $\frac{n}{2}$ so $m \in \{1, 2, \ldots, \frac{n}{2}\}$ (also in bytes). Measurement results of the 100 MB case are plotted in figure 2(b). Tests with data inputs of 1 MB and 10 MB showed that the number of collisions, the search and the encryption times are proportional to the data size. In our technical report [6] more experimental data is provided. All tests were carried out on a Pentium IV 2.4 MHz with 512 MB memory.

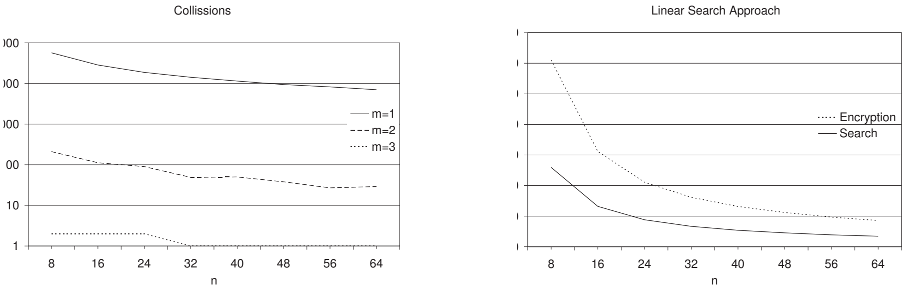
For the search query a word guaranteed to be in at least one location was chosen. The search engine does not stop when one occurrence is found; all the text is scanned for each query.

4.2 Experiments with the Tree Search Prototype

For the tree search prototype the only configurable parameters are m and the data size. The block length n depends on the tag names and values. Encryption tests are carried out on the same XML documents as in the linear prototype. In this case m is relative to n; $m \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. The encryption times for the 1 MB, 10 MB and the 100 MB files were 21.5, 188 and 1195 s and did not depend on m.

Search tests were carried out with a fixed $m = 0.5$ because m does not seem to have much influence. Some queries are shown in table 1. Also the number of

[1] http://www.xml-benchmark.org



(a) Number of Measured Collisions

(b) Encryption and Search Search Times

Fig. 2. Measurement Results of Linear Search Prototype for the 100 MB Case

elements in the result is shown for each query. All three files have approximately the same tree depth but have different branch factors (average number of sub children per element).

Table 1. Search Times Calculated for Search Queries with Different Depth and Branch Factor

t (ms) 1 MB	t (ms) 10 MB	t (ms) 100 MB	query	count 1 MB	count 10 MB	count 100 MB
1281	1506	1285	/site	1	1	1
1266	1380	1321	/site/regions	1	1	1
1358	1435	1342	/site/regions/asia	1	1	1
1409	1687	2464	/site/regions/asia/item	20	200	2000
1518	2030	4135	/site/regions/asia/item/description	20	200	2000
1376	1591	2442	/site/regions/africa/item/description	5	55	550
1448	2777	9059	/site/regions/europe/item/description	60	600	6000
1455	2098	4577	/site/regions/australia/item/description	22	220	2200
1654	3226	13672	/site/regions/namerica/item/description	100	1000	10000
1336	1817	3028	/site/regions/samerica/item/description	10	100	1000
1398	2382	18530	//*	21048	206130	2048180
3639	21775	191899	//item	217	2175	21750

5 Analysis of the Results

First we will analyse the results of the individual experiments in the first two subsections. In subsection 5.3 we will compare the linear search approach with the tree search approach.

5.1 Results from the Linear Search Approach

From the linear search prototype we can conclude the following:

- As expected the larger the dataset the larger the encryption and search times. Encryption and search times grow linear in the size of the dataset. Therefore the protocol does not scale well and can only be used for reasonable small databases.
- The larger n is the shorter the encryption and search times gets (figure 2(b)). This can be explained by looking at the number of blocks. The larger n is the fewer blocks there are. For each block a fixed number of steps is taken. Most of these steps do not depend on the length of the blocks. Therefore less time is needed for the whole database.
- Searching is faster than encryption, because fewer operations have to be calculated for each block.
- The larger n is the fewer collisions occur (figure 2(a)). This can also be explained by the fewer blocks.
- For a fixed value of n the encryption and search times hardly depend on the value of m.
- Collisions can be avoided by choosing a sufficiently large value of m. The largest value for $m = \frac{n}{2}$ which is also the most optimal one. But also for $m > 2$ the number of collisions is negligible.

5.2 Results from the Tree Search Approach

From the tree search prototype we can conclude that:

- The encryption time is linear in the size of the input.
- The search time depends both on the structure of the XML document and the search query. The search time is of order $O(p)$ where p is the number of elements to be read. For queries without // this comes down to $O(bd)$ where b is the branch factor (the average number of sub elements) and d is the depth in the tree where the answer is found.

5.3 Benefits of using Tree Structure

From the experiments with the linear search method we know that the encryption time depends on the block size. Therefore, to make a fair comparison between the linear text search and the tree search, we have to take into account the block size of the tree search method. We analysed the XML documents and found the data shown in table 2.

Comparison of the encryption speed in the tree search case (with an average block size of around 18) with the linear case, shows that the tree encryption is slightly faster than in the linear case. The reason for this is that there is no need to encrypt the close tag.

The major benefit of using the tree structure is the increase in search speed. Only a small part of the whole tree has to be searched. Because the search time totally depends on the data and the query, a straight comparison between the linear and the tree case is impossible. However, linear search is of order $O(n) = O(b^d)$, whereas tree searching is of order $O(bd)$.

6 Conclusions

We have implemented a prototype for the theory described in [1]. We showed that the search complexity is linear in the size of the text. We have defined a new protocol for semi-structured XML data that exploits the tree structure. Experiments with the implementations of both protocols showed that the encryption speed remains linear in the size of the input, but that a major improvement in the search speed can be achieved.

References

1. Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy*, pages 44–55, 2000. http://citeseer.nj.nec.com/song00practical.html.
2. Torsten Grust. Accelerating xpath location steps. In *Proceedings of the 21st ACM International Conference on Management of Data (SIGMOD 2002)*, pages 109–120. ACM Press, Madison, Wisconsin, USA, June 2002. http://www.informatik.uni-konstanz.de/∼grust/files/xpath-accel.pdf.
3. Torsten Grust, Maurice van Keulen, and Jens Teubner. Staircase join: Teach a relational dbms to watch its (axis) steps. In *Proceedings of the 29th Int'l Conference on Very Large Databases (VLDB 2003)*, Berlin, Germany, Sep 2003. http://citeseer.nj.nec.com/593676.html.
4. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, October 1996. http://www.cacr.math.uwaterloo.ca/hac/.
5. A. Schmidt, F. Waas, M. Kersten, D. Florescu, I. Manolescu, M. Carey, and R. Busse. The xml benchmark project. Technical Report INS-R0103, CWI, April 2001. http://citeseer.ist.psu.edu/schmidt01xml.html.
6. R. Brinkman, L. Feng, S. Etalle, P. H. Hartel, and W. Jonker. Experimenting with linear search in encrypted data. Technical report TR-CTIT-03-43, Centre for Telematics and Information Technology, Univ. of Twente, The Netherlands, Sep 2003. http://www.ub.utwente.nl/webdocs/ctit/1/000000d9.pdf.

Table 2. Block Sizes

data size	avg tag length	standard deviation	avg text size	standard deviation	avg all blocks	standard deviation
1 MB	9.8	3.4	28.0	70	18.2	48
10 MB	9.8	3.4	28.6	70	18.4	48
100 MB	9.8	3.4	28.9	70	18.6	49

A Generalized Policy Support System and Its Hierarchy Semantics

Yibing Kong, Janusz R. Getta, Ping Yu, and Jennifer Seberry

School of Information Technology and Computer Science,
University of Wollongong,
Wollongong, NSW, Australia
{yk18, jrg, ping, jennie}@uow.edu.au

Abstract. One common characteristic of many *Policy Support Systems* ($\mathcal{PSS}$s) is their dependency on the concept of *hierarchy*. Hierarchy does not need to be limited to a hierarchy of roles (subject centric) as in traditional Role-Based Access Control (RBAC). Instead, it can be applied to other aspects of $\mathcal{PSS}$ such as object, environment, purpose and so on. In this paper, we propose a new generalized model for $\mathcal{PSS}$. The model unifies Generalized Role-Based Access Control (GRBAC) and Enterprise Privacy Practices (E-P3P) policy support systems and generalizes their hierarchy semantics.

Keywords: Access Control, Hierarchy, Hierarchy Semantics

1 Introduction

Organizations must enforce data protection policies to secure data collected and generated during their daily operational procedures. At a conceptual level, data protection polices are expressed as sequences of statements written in a natural language. In the past, the enforcement of data protection policies was based on manual work or inflexible mechanisms such as Discretionary Access Control (DAC), Mandatory Access Control (MAC) models. The emerge of Role-Based Access Control (RBAC) model improved efficiency of data protection. Unfortunately, RBAC model is still not expressive enough to efficiently and effectively enforce more sophisticated data protection policies in an organization. Recently more powerful systems have appeared; Generalized Role-Based Access Control (GRBAC) [1] and Enterprise Privacy Practices (E-P3P) [2] are representatives of such systems.

GRBAC, proposed by Moyer and Ahamad in [1], is an extension of traditional RBAC. It generalizes the classical concept of *role* through the new concepts such as *subject role*, *object role* and *environment role*. These concepts are used to structure the subjects (users), objects (data) and environments (conditions). With these new types of roles, GRBAC is capable of creating rich access control policies. GRBAC provides an algorithm to enforce the access control policies defined in the model. E-P3P, developed by Ashley *et al.* in [2], has a well-defined privacy architecture and semantics. It enables an organization to express its privacy policies in E-P3P format and to enforce the policies automatically. We shall call these two systems as *Policy Support Systems* ($\mathcal{PSS}$s) because they are designed for expressing and enforcing data protection policies.

A *hierarchy* is a partial order on a set of elements that defines a seniority relationship between elements [3]. Hierarchy is not a new concept, it has been extensively studied in the past. *Role hierarchy* [4–7] in role-based access control and *hierarchy* in Flexible Authorization Framework (FAF) [8] are examples of such studies. Hierarchy semantics is an inseparable part of hierarchy that defines rules of authorization propagation. Various hierarchy semantics was defined in GRBAC and E-P3P. However, the hierarchy semantics defined is either incomplete or incorrect. In this paper, we propose a generalized $\mathcal{PSS}$ model that covers GRBAC and E-P3P. Based on this model, the hierarchy semantics of GRBAC and E-P3P is analyzed. We propose new hierarchy semantics to solve the problems encountered in GRBAC and E-P3P.

The organization of the rest of the paper is as follows. Section 2 introduces the concept of hierarchy. A generalized $\mathcal{PSS}$ is proposed in section 3 and hierarchy semantics of GRBAC and E-P3P is analyzed in section 4. Section 5 presents new hierarchy semantics. Finally, in section 6, we conclude the paper and outline the plans of future research.

2 Definition of Hierarchy

A mathematical structure called *hierarchy* was defined by Jajodia *et al.* in [8] as a triple $(X, Y, \leq)$, where X is the set of *primitive entities*, e.g. a user, an object; Y is the set of *categories*, e.g. a group, an object type; $\leq$ is a partial order on $(X \cup Y)$ such that each $x \in X$ is a *minimal element* of $(X \cup Y)$; an element $x \in X$ is said to be minimal iff there are no elements below it in the hierarchy, that is iff $\forall y \in (X \cup Y) : y \leq x \Rightarrow y = x$. This definition is rich enough to capture all hierarchy structures presented in [1, 2]. We simplify this definition of hierarchy to a two-entry tuple $(Y, \leq)$, i.e. $H = (Y, \leq)$ where:

- Y is the set of categories, such that a primitive entity is treated as a category of itself, called *primitive category*. A primitive category contains one primitive entity and the name of the category is the same as the name of the primitive entity. For example a primitive entity *James Bond* belongs to a primitive category called `James Bond`.
- $\leq$ is a partial order on Y such that each primitive category in Y is a minimal element of Y.

In addition, we define the following two binary relations over Y. The first binary relation $<$ describes *descendant-ancestor* relationship between the elements in Y. If $y_i, y_j \in Y$ and $y_i < y_j$, y_j is said to be the ancestor of y_i; y_i is said to be the descendant of y_j. The relation $y_i < y_j$ is interpreted as y_i is *in the category of* y_j. For an element y_i, a set of all its ancestors is defined as $Aset_{y_i} = \{y_k : y_k \in Y \text{ and } y_i < y_k\}$; a set of all its descendants is $Dset_{y_i} = \{y_k : y_k \in Y \text{ and } y_k < y_i\}$. The second binary relation $<^C$ describes *child-parent* relationship between elements in Y. If $y_i, y_j \in Y$ and $y_i <^C y_j$, y_j is said to be the parent of y_i; y_i is said to be the child of y_j. For an element y_i, a set of all its parents is defined as $Pset_{y_i} = \{y_k : y_k \in Y \text{ and } y_i <^C y_k\}$; a set of all its children is $Cset_{y_i} = \{y_k : y_k \in Y \text{ and } y_k <^C y_i\}$.

Some of the hierarchies used in practice are listed as follows. *Subject hierarchy* $GH = (G, \leq_G)$ where G is a set of groups (roles), $\leq_G$ defines hierarchy relationships between groups in G. *Object hierarchy* $TH = (T, \leq_T)$ where T is a set of types, $\leq_T$ defines hierarchy relationships between types in T. *Environment hierarchy* $EH = (E, \leq_E)$ where E is a set of environments, $\leq_E$ defines hierarchy relationships between environments in E. *Purpose hierarchy* $PH = (P, \leq_P)$ where P is a set of purposes, $\leq_P$ defines hierarchy relationships between purposes in P.

3 A Generalized Model of a Policy Support System

It is common for a $\mathcal{PSS}$ to define more than one hierarchies. For example, GRBAC [1] defines GH, TH and EH hierarchies and E-P3P [2] defines GH, TH and PH hierarchies. Furthermore these systems define some sets of elements, such as the set of actions, the set of obligations, the set of authorization types etc. In this section, we define a generalized model which unifies GRBAC and E-P3P.

A generalized *Policy Support System* $\mathcal{PSS} = (\mathcal{H}, \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$, where:

- $\mathcal{H}$ denotes a set of n hierarchies $H_1, ..., H_n$, $n \geq 1$.
- $\mathcal{S}$ is a set of m sets $S_1, ..., S_m$, where $m \geq 0$. The sets defined in $\mathcal{S}$ provide additional restrictions on $\mathcal{PSS}$; for example the sets of obligations and conditions in E-P3P system are instances of such sets. $\mathcal{S}$ is optional and its existence depends on the designer of a $\mathcal{PSS}$, e.g. in GRBAC $\mathcal{S}$ is absent.
- $\mathcal{A}$ is a set of actions to be performed on data.
- $\mathcal{R}$ is a set of authorization types (rulings). $R = \{+, -, \oplus, \ominus, \odot, \otimes\}$, where $+$ means *positive authorization*; $-$ means *negative authorization*; $\oplus$ means *implicit positive authorization*; $\ominus$ means *implicit negative authorization*; $\odot$ means *authorization pending* and $\otimes$ means *no authorization*. $+$ and $-$ are used for explicit authorization assignment through policy rules; $\oplus$ and $\ominus$ are used for authorization propagation; $\odot$ is used for conflicts resolution; $\otimes$ is used when none of the above five rulings is applicable.
- $\mathcal{P}$ is a set of precedences over policy rules. $\mathcal{P} = \mathbb{Z}$, i.e. $\mathcal{P}$ is the set of integers that determines precedence orders over a set of policy rules; the greater number denotes the higher precedence. $\mathcal{P}$ is also optional; the absence of $\mathcal{P}$ means that all policy rules have the same precedence.

The elements of the above five parts are used as basic units to form policy rules. The collection of all the policy rules in a $\mathcal{PSS}$ is a policy rule set, denoted as Γ. A *policy rule* $\gamma \in \Gamma$ is a tuple $(\gamma_{H_1}, ..., \gamma_{H_n}, \gamma_{S_1}, ..., \gamma_{S_m}, \gamma_{\mathcal{A}}, \gamma_{\mathcal{R}}, \gamma_{\mathcal{P}})$ where

- $\gamma_{H_i} \in H_i$ or $\gamma_{H_i} = null$ (i.e. nothing is specified for γ_{H_i}), where $n \geq i \geq 1$.
- $\gamma_{S_i} \in S_i$ or $\gamma_{S_i} = null$, where $m \geq i \geq 0$.
- $\gamma_{\mathcal{A}} \in \mathcal{A}$ is the action entry that specifies the action to be performed.
- $\gamma_{\mathcal{R}} \in \{+, -\}$ is the ruling entry that specifies either positive or negative authorization.
- $\gamma_{\mathcal{P}} \in \mathcal{P}$ is the precedence of γ.

It is possible to show that the model defined above "includes" GRBAC [1] and E-P3P [2]. A GRBAC system is a triple $(\mathcal{H}, \mathcal{A}, \mathcal{R})$, where $\mathcal{H} = \{GH, TH, EH\}$ (GH is subject hierarchy, TH is object hierarchy and EH is environment hierarchy). A GRBAC policy rule [1] is a tuple $(S, O, E, op,$ *permission bit*$)$, where $S \in GH, O \in TH, E \in EH, op \in \mathcal{A}$ and *permission bit* $\in \{+, -\}$. There is no precedence over GRBAC policy rules, hence the set $\mathcal{P}$ is absent. An E-P3P system is a tuple $(\mathcal{H}, \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$, where $\mathcal{H} = \{GH, TH, PH\}, \mathcal{S} = \{O, C\}$ (GH is subject hierarchy, TH is object hierarchy, PH is purpose hierarchy, O is the set of obligations and C is the set of conditions). An E-P3P policy rule [2] is a tuple $(i, t, p, u, r, a, \bar{o}, \bar{c})$, inside which $i \in \mathcal{P}, t \in TH, p \in PH, u \in GH, r \in \{+, -\}, a \in \mathcal{A}, \bar{o} \in O$ and $\bar{c} \in C$.

According to the definition of a policy rule, an *access request* α can be expressed as $\alpha = (\alpha_{H_1}, ..., \alpha_{H_n}, \alpha_{S_1}, ..., \alpha_{S_m}, \alpha_{\mathcal{A}})$ where $\alpha_{H_i} \in H_i$ or $\alpha_{H_i} = null$, $n \geq i \geq 1$; $\alpha_{S_i} \in S_i$ or $\alpha_{S_i} = null$, $m \geq i \geq 0$; $\alpha_{\mathcal{A}} \in \mathcal{A}$. A set of policy rules $\varGamma_\alpha$ is used to validate α, where $\varGamma_\alpha \subseteq \varGamma$. All policy rules in $\varGamma_\alpha$ are called *matching rules* of α. Matching rules must satisfy the following properties:

- $\forall \gamma \in \varGamma_\alpha, \alpha_{H_i} \leq \gamma_{H_i}$, where $n \geq i \geq 1$.
- $\forall \gamma \in \varGamma_\alpha, \gamma_{S_i} = \alpha_{S_i}$, where $m \geq i \geq 0$.
- $\forall \gamma \in \varGamma_\alpha, \gamma_{\mathcal{A}} = \alpha_{\mathcal{A}}$.

The validation of α in $\varGamma_\alpha$ consists of n sub-validations from α_{H_1} to α_{H_n}. That is, $\forall \alpha_{H_i} \in \alpha$ where $n \geq i \geq 1$, α_{H_i} needs to be validated according to the matching rule set $\varGamma_\alpha$ whether α_{H_i} is authorized for $\alpha_{\mathcal{A}}$. If and only if all of these hierarchy elements are authorized for $\alpha_{\mathcal{A}}$, α is granted.

$\forall \gamma \in \varGamma_\alpha$, where $\gamma = (\gamma_{H_1}, ..., \gamma_{H_n}, \gamma_{S_1}, ..., \gamma_{S_m}, \gamma_{\mathcal{A}}, \gamma_{\mathcal{R}}, \gamma_{\mathcal{P}})$, we can assign a tuple $(\gamma_{\mathcal{R}}, \gamma_{\mathcal{P}})$ to $\gamma_{H_1}, ..., \gamma_{H_n}$. An *authorization* of an element $y \in H$ is a tuple $(ruling, precedence)$ inside which $ruling \in \mathcal{R}, precedence \in \mathcal{P}$; it is denoted as A_y^i where i is the index of the authorization. Due to the optional property of $\mathcal{P}$, the precedence entry of an authorization is also optional. The ruling entry of A_y^i is denoted as $A_y^i.ruling$ and the precedence entry of A_y^i is denoted as $A_y^i.precedence$. We call an authorization explicitly defined by policy rules *explicit authorization*; obviously for an explicit authorization A, $A.ruling \in \{+, -\}$. Authorization may also be derived by hierarchy semantics, we call a derived authorization *implicit authorization*; for an implicit authorization A, $A.ruling \in \{\oplus, \ominus\}$. If an explicit authorization $A_{y_i}^1$ of $y_i \in H$ propagates to $y_j \in H$, then $A_{y_i}^1$ is converted to an implicit authorization $A_{y_j}^1$ by the following processes: if $A_{y_i}^1.ruling = +$, then $A_{y_j}^1.ruling = \oplus$; if $A_{y_i}^1.ruling = -$, then $A_{y_j}^1.ruling = \ominus$; $A_{y_j}^1.precedence = A_{y_i}^1.precedence$.

Here is an example of assigning explicit authorizations, assume $\varGamma_\alpha = \{\gamma_1, \gamma_2\}$; $\gamma_1 = (..., \gamma_{1H_i}, ..., read, +, 1)$, $\gamma_2 = (..., \gamma_{2H_i}, ..., read, -, 2)$, where $\gamma_{1H_i} = \gamma_{2H_i} = y \in H_i$; then $A_y^1 = (+, 1)$, $A_y^2 = (-, 2)$. Two authorizations A_y^i, A_y^j are *inequable* if either $A_y^i.ruling \neq A_y^j.ruling$ or $A_y^i.precedence \neq A_y^j.precedence$ holds. When an element of a hierarchy has more than one inequable authorizations, conflicts arise. Our system provides methods of conflicts resolution, called *authorization precedence policy*. In our system, conflicts can be solved either *manually* or *automatically*. For a hierarchy $H_i = (Y_i, \leq_i)$, the *System Security Officer* (SSO) defines a *manual resolution set* $MR_i \subseteq Y_i$. If an element $y \in MR_i$ has authorization conflicts, we assign y with the authorization $(\odot,)$. In this case, the decision of the access request α that causes the conflicts will be pending until the conflicts are manually solved by SSO. If an element $y \in Y_i \setminus MR_i$ has authorization conflicts, these conflicts are solved automatically by the following rules.

- $\odot$ authorizations are with the highest precedence; $\otimes$ authorizations are with the lowest precedence. There are no maximum and minimum integers in $\mathbb{Z}$, hence the precedence entries of these two types of authorizations are absent. An authorization with higher precedence overrides authorizations with lower precedences.
- If the precedences are the same, *denies-take-precedence* will apply. The priority order is: $-, \ominus, +, \oplus$. An authorization with higher priority order overrides authorizations with lower priority orders.

Our authorization precedence policy is very flexible. Authorization pending gives SSO opportunities to review authorization conflicts. By doing that, SSO may find bugs of Γ and give user better response. After we perform all authorization derivations and conflicts resolutions on H_i, H_i's *final authorization state* is obtained, where $\forall y \in H_i$, y has a *final authorization* FA_y that is not conflicting. The result of the sub-validation of α_{H_i} is $FA_{\alpha_{H_i}}.ruling$. The decision of α is processed as follows.

- If $\forall FA_{\alpha_{H_i}}.ruling = +$ (or $\oplus$) where $n \geq i \geq 1$, then α is approved.
- If $\exists FA_{\alpha_{H_i}}.ruling = -$ (or $\ominus$ or $\otimes$), then α is denied.
- If α is not denied and $\exists FA_{\alpha_{H_i}}.ruling = \odot$, then α is pending.

4 Hierarchy Semantics of GRBAC and E-P3P

Hierarchy semantics defines rules of authorization propagation. In this section, the hierarchy semantics of GRBAC and E-P3P is depicted.

4.1 Hierarchy Semantics of GRBAC

The hierarchy semantics in GRBAC [1] is defined by *permission inheritance*. There are three types of permission inheritances: *standard*, *strict* and *lenient*. Suppose we have an access request $\alpha = (\alpha_{GH}, \alpha_{TH}, \alpha_{EH}, \alpha_{\mathcal{A}})$, where $\alpha_{GH} = y_4 \in GH$, denoted as $GH.y_4$, $\alpha_{TH} = TH.y_5$ and $\alpha_{EH} = EH.y_2$. Here we utilize the validation of $GH.y_4$ to illustrate the semantics of the three types of permission inheritances.

- *Standard permission inheritance*:
  If $\exists y_i \in Aset_{GH.y_4} \cup \{GH.y_4\}$ such that $FA_{y_i}.ruling = +$ and $\neg\exists y_j \in Aset_{GH.y_4} \cup \{GH.y_4\}$ such that $FA_{y_j}.ruling = -$, then $GH.y_4$ is authorized for $\alpha_{\mathcal{A}}$. Otherwise, it is not authorized for $\alpha_{\mathcal{A}}$.
- *Lenient permission inheritance*:
  If $\exists y_i \in Aset_{GH.y_4} \cup \{GH.y_4\}$ such that $FA_{y_i}.ruling = +$, then $GH.y_4$ is authorized for $\alpha_{\mathcal{A}}$. Otherwise, it is not authorized for $\alpha_{\mathcal{A}}$.
- *Strict permission inheritance*:
  If $\forall y_i \in Aset_{GH.y_4} \cup \{GH.y_4\}$ such that $FA_{y_i}.ruling = +$, then $GH.y_4$ is authorized for $\alpha_{\mathcal{A}}$. Otherwise, it is not authorized for $\alpha_{\mathcal{A}}$.

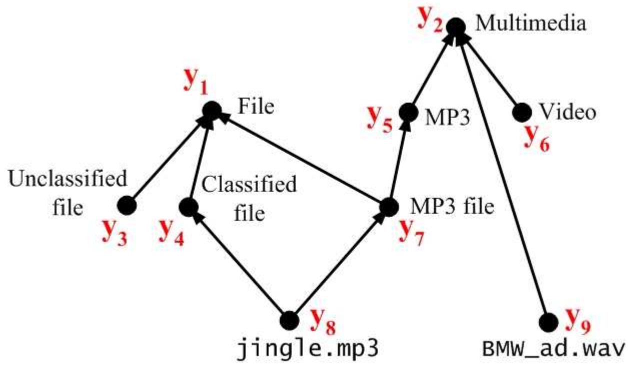


Fig. 1. Example object hierarchy TH

An example below illustrates the hierarchy semantics of GRBAC. Consider the object hierarchy TH shown in figure 1; there are 9 elements in TH, among which `jingle.mp3` (y_8) is a primitive category that is a child of *classified file* (y_4) and *MP3 file* (y_7). Now there is an access request α from user u (we assume u is y_6 in GH) to read `jingle.mp3` ($TH.y_8$) under environment e (we assume e is y_1 in EH), i.e. $\alpha = (GH.y_6, TH.y_8, EH.y_1, read)$. The policy rule set $\Gamma = \{\gamma_1, \gamma_2\}$; $\gamma_1 = (GH.y_6, TH.y_4, EH.y_1, read, +)$, $\gamma_2 = (GH.y_6, TH.y_7, EH.y_1, read, +)$. Obviously, the matching rule set $\Gamma_\alpha = \Gamma$. According to GRBAC hierarchy semantics, we can derive that u can read `jingle.mp3` if standard or lenient permission inheritance is applied; u cannot read it if strict permission inheritance is applied.

The original intention of strict permission inheritance is to restrict accesses to the elements in the category of sensitive/vulnerable categories defined by SSO. GRBAC's strict permission inheritance is trying to fulfill this intention. However this definition is too strict to be practical. In the example above, user u is explicitly authorized to read both *classified file* and *MP3 file* and there is no policy rule disallow these accesses (as shown in the example Γ above). In this case, even if we are very strict, user u should have read access to `jingle.mp3`, which is a child of *classified file* and *MP3 file*. GRBAC will deny this access because GRBAC's strict permission inheritance requires the requested element and all its ancestors to be explicitly authorized for the access; obviously this is too strict. If a system deploys this strict permission inheritance, few access requests can be granted.

4.2 Hierarchy Semantics of E-P3P

In E-P3P, an access request α is processed in the following two steps [2]. The first step creates a set of preliminary authorization rules PA; PA is a rule set defined as the union of Γ and $D\Gamma$, where $D\Gamma$ contains all rules derived from Γ by using the hierarchy semantics defined in this system. The second step processes access request α according to PA.

The hierarchy semantics of E-P3P is defined as follows [2]:

- Down-inheritance: For each rule $(i, t, p, u, r, a, \overline{o}, \overline{c}) \in PA$, for every (t', p', u') such that $t' \leq_T t$, $p' \leq_P p$, and $u' \leq_G u$, a tuple $(i, t', p', u', r, a, \overline{o}, \overline{c})$ is added to PA.
- Up-inheritance of deny (negative authorization): For each rule $(i, t, p, u, -, a, \overline{o}, \overline{c}) \in PA$, for every (t', p', u') such that $t \leq_T t'$, $p \leq_P p'$, and $u \leq_G u'$, a tuple $(i, t', p', u' -, a, \overline{o}, \overline{c})$ is added to PA.

In this system, if contradicting policy rules coexist, *denies-take-precedence* will be applied to remove contradicting policy rules with lower precedences from PA.

We can identify two problems existing in this system. The first problem is that the concept of *permission inheritance* is omitted. As a consequence, the system is not flexible in practice. For example, it provides no mechanism for enforcing *strict permission inheritance*. The second problem is that the definition of *up-inheritance of deny* is reasonless. The first problem is apparent; here we will give an example that reveals the second problem. Following the semantics of *up-inheritance of deny*, some reasonable requests from users are denied. Let us consider the following scenario. There is a primitive category `BMW_ad.wav` ($TH.y_9$ in figure 1) in the category of *multimedia* ($TH.y_2$ in figure 1), besides we assume that $GH.y_6$ is user u and $PH.y_3$ is a purpose. There are two policy rules in $\mathit{\Gamma}$, i.e. $\mathit{\Gamma} = \{\gamma_1, \gamma_2\}$; $\gamma_1 = (1, TH.y_7, PH.y_3, GH.y_6, -, read, null, null)$, $\gamma_2 = (1, TH.y_2, PH.y_3, GH.y_6, +, read, null, null)$. Now there is an access request from user u: $\alpha = (TH.y_9, PH.y_3, GH.y_6, read, null, null)$; i.e. user u requests to read `BMW_ad.wav` for the purpose of $PH.y_3$ with no specified condition and obligation. Because $TH.y_7 \leq_T TH.y_2$ (see figure 1), $PH.y_3 \leq_P PH.y_3$ and $GH.y_6 \leq_G GH.y_6$, following up-inheritance of deny, there will be a policy rule γ_3 derived from γ_1: $\gamma_3 = (1, TH.y_2, PH.y_3, GH.y_6, -, read, null, null)$, that is user u is not allowed to read *multimedia* for the purpose of $PH.y_3$. Then $PA = \mathit{\Gamma} \cup \{\gamma_3\}$, i.e. $PA = \{\gamma_1, \gamma_2, \gamma_3\}$ (here we skip other derived rules). The two policy rules γ_2 and γ_3 are contradicting policy rules. Because of denies-take-precedence, the rule γ_2 will be removed from PA, now $PA = \{\gamma_1, \gamma_3\}$. The system will validate α according to $PA = \{\gamma_1, \gamma_3\}$, hence according to γ_3, user u's request α is denied.

5 Solution

This section presents the hierarchy semantics defined in our generalized $\mathcal{PSS}$. The semantics described below eliminates the problems mentioned in section 4 and extends the hierarchy semantics of GRBAC and E-P3P.

5.1 Hierarchy Semantics

Our interpretation of the concept of hierarchy is such that the relationship between a descendant element and its ancestor element is *in the category of*. The common rationale is that when an authorization is applied on an ancestor element (superior category), this authorization may also be applied to its descendant elements (inferior categories) implicitly. As a consequence, authorizations propagate downwards ($\odot$ and $\otimes$ authorizations do not propagate; the term *authorization* in this sub-section denotes authorizations other than $\odot$ and $\otimes$ authorizations). We only consider authorization propagations between parents and children here; any complex authorization propagation is an aggregation of such simple propagations. In a hierarchy, two different types of elements must be clearly distinguished. *Pure element* is an element that has only one parent; *hybrid element* is an element that has more than one parents. Based on these two types of elements, two different situations of down-propagation of authorizations can be identified.

- If a child is a pure element, all authorizations of its parent propagate down.
- If a child is a hybrid element, the hierarchy semantics is complicated. There are many options that represent different strictness of authorization propagation. These options are listed as follows.
  (a) *Strict down-propagation*: SSO defines a combination of elements called *Strict Combination* (SC). For a child y, if $\exists y_j \in Pset_y$ such that $y_j \in SC$, then the authorization propagation from y's parents to y will follow the semantics of strict down-propagation. We define a set $SC_y = \{y_i : y_i \in Pset_y$ and $y_i \in SC\}$. The semantics of strict down-propagation is as follows.
     i. All (implicit) negative authorizations of elements in $Pset_y \backslash SC_y$ propagate down to y; all other authorizations of elements in $Pset_y$ propagate down to y iff $\forall y_i \in SC_y$ such that $FA_{y_i}.ruling = +$ (or $\oplus$).
  (b) *Lenient down-propagation*: SSO defines a combination of elements called *Lenient Combination* (LC). For a child y, if $\exists y_j \in Pset_y$ such that $y_j \in LC$ and $\neg\exists y_k \in Pset_y$ such that $y_k \in SC$ (for security concern, strict down-propagation overrides lenient down-propagation), then the authorization propagation from y's parents to y will follow the semantics of lenient down-propagation. We define a set $LC_y = \{y_i : y_i \in Pset_y$ and $y_i \in LC\}$. The semantics of lenient down-propagation is as follows.
     i. If $\neg\exists y_i \in LC_y$ such that $FA_{y_i}.ruling = +$ (or $\oplus$), all authorizations of elements in $Pset_y$ propagate down to y.
     ii. If $\exists y_i \in LC_y$ such that $FA_{y_i}.ruling = +$ (or $\oplus$), all authorizations of elements in $\{y_k : y_k \in Pset_y$ and $FA_{y_k}.ruling = +$ (or $\oplus$)$\}$ propagate down to y.
  (c) *Standard down-propagation*: For a child y, if $\neg\exists y_j \in Pset_y$ such that $y_j \in LC$ and $\neg\exists y_k \in Pset_v$ such that $y_k \in SC$, then the authorization propagation from y's parents to y will follow the semantics of standard down-propagation. The semantics of standard down-propagation is as follows.
     i. All authorizations of y's parents propagate down to y.

The semantics described above generalizes the hierarchy semantics in GRBAC and E-P3P.The strict permission inheritance defined in GRBAC (section 4.1) is a special case of our definition of strict down-propagation where $\forall y_i \in Pset_y, y_i \in SC$. The lenient permission inheritance defined in GRBAC (section 4.1) is a special case of our definition of lenient down-propagation where $\forall y_i \in Pset_y, y_i \in LC$. The proposed hierarchy semantics also eliminates the questionable semantics in GRBAC and E-P3P. If our strict down-propagation is applied in the examples shown in section 4.1 and section 4.2, the reasonable access requests will be approved. The semantics of up-inheritance of deny (section 4.2) is incorrect; hence in our hierarchy semantics it is not included.

5.2 Scenarios of the Use of Hierarchy Semantics

This section shows some examples of using the hierarchy semantics defined in this paper. In these examples, we assume authorization conflicts are resolved automatically. Due to limited space, the examples of down-propagation to a pure element and standard down-propagation are not included in this paper.

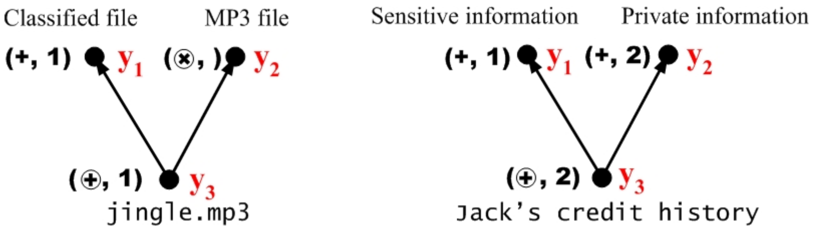


Fig. 2. Example strict down-propagation of object hierarchy

Figure 2 shows two examples of strict down-propagation. In the first example, a primitive category `jingle.mp3` (y_3) enters two categories: *classified file* (y_1) and *MP3 file* (y_2). In this case, SSO wants to be strict to accesses to elements entering category y_1. SSO defines $SC = \{y_1\}$. Because $FA_{y_1}.ruling = +$, FA_{y_1} propagates down to y_3: $A^1_{y_3} = (\oplus, 1)$. $A^1_{y_3}$ is the only authorization that y_3 has, hence $FA_{y_3} = A^1_{y_3}$. In the second example, a primitive category `Jack's credit history` (y_3) enters two categories: *sensitive information* (y_1) and *private information* (y_2). SSO wants to be strict to accesses to elements entering y_1 or y_2. SSO defines $SC = \{y_1, y_2\}$. Because $FA_{y_1}.ruling = +$ and $FA_{y_2}.ruling = +$, FA_{y_1} and FA_{y_2} propagate down to y_3. After conflict resolution, $FA_{y_3} = (\oplus, 2)$.

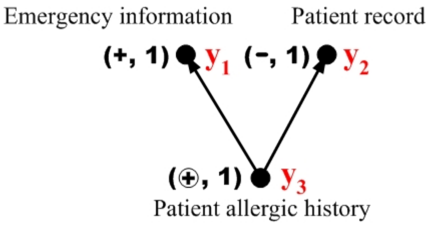


Fig. 3. Example lenient down-propagation of object hierarchy

An example of lenient down-propagation is shown in figure 3. SSO wants to be lenient to those who are allowed to access *emergency information* (y_1), because *emergency information* is often related to vital event. SSO defines $LC = \{y_1\}$ and we assume $SC = \emptyset$. Then even if the other parent y_2 of *patient allergic history* (y_3) is denied access, y_3 is still accessible to those who are allowed to access y_1.

6 Conclusion and Future Work

$\mathcal{PSS}$s are capable of expressing and enforcing rich data protection policies. GRBAC and E-P3P are representatives of such systems. In GRBAC and E-P3P, hierarchy is an important and widely used concept. Being an inseparable part of hierarchy, hierarchy semantics defines rules of authorization propagation. In this paper, we have proposed a generalized $\mathcal{PSS}$ that covers GRBAC and E-P3P. Based on this generalized $\mathcal{PSS}$, we analyze the hierarchy semantics used in GRBAC and E-P3P. We point out errors and limitations of GRBAC and E-P3P hierarchy semantics. Finally, we present new hierarchy semantics to address the problems discovered.

In the future, the following research work interests us:

- finding more useful hierarchy semantics.
- investigating efficient access request processing mechanisms.
- reviewing other related work such as access control for XML document etc.

References

1. Moyer, M.J., Ahamad, M.: Generalized role-based access control. In: Proceedings of 21st International Conference on Distributed Computing Systems. (2001) 391–398
2. Ashley, P., Hada, S., Karjoth, G., Schunter, M.: E-P3P privacy policies and privacy authorization. In: Proceeding of the ACM workshop on Privacy in the Electronic Society, ACM Press (2002) 103–109
3. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC) **4** (2001) 224–274
4. Moffett, J.D.: Control principles and role hierarchies. In: Proceedings of the third ACM workshop on Role-based access control, ACM Press (1998) 63–69
5. Sandhu, R.: Role activation hierarchies. In: Proceedings of the third ACM workshop on Role-based access control, ACM Press (1998) 33–40
6. Joshi, J.B.D., Bertino, E., Ghafoor, A.: Hybrid role hierarchy for generalized temporal role based access control model. In: Proceedings of 26th Annual International Computer Software and Applications Conference. (2002) 951–956
7. Moffett, J.D., Lupu, E.C.: The uses of role hierarchies in access control. In: Proceedings of the fourth ACM workshop on Role-based access control, ACM Press (1999) 153–160
8. Jajodia, S., Samarati, P., Sapino, M.L., Subrahmanian, V.S.: Flexible support for multiple access control policies. ACM Transactions on Database Systems (TODS) **26** (2001) 214–260

The Impact of Virus Attack Announcements on the Market Value of Firms

Anat Hovav[1] and John D'Arcy[1]

[1] Temple University, MIS Department, Fox School of Business and Management,
1810 N. 13th Street, Philadelphia, PA 19122 USA
anat.hovav@temple.edu
jdarcy@temple.edu

Abstract. The increase in security breaches in the last few years and the need to insure information assets has created an intensified interest in information security and risk within organizations. However, very little is known of the financial impact and the risk associated with the various types of security breaches. This article reports the impact of virus attack announcements on the market value of affected companies over a period of 15 years. The study was conducted using event study methodology. The results show that in general the market does not penalize companies that experience such an attack.

Keywords: information systems security, security breach, computer virus, event study

1 Introduction

Information Systems (IS) risk is a top concern for organizations [33]. These concerns are due to the fact that the consequence of a security breach can be detrimental to a company's financial performance [13]. Thus, security strategies revolve around the act of a security breach (or an attempt at one) and the need to minimize the financial loss resulting from such a breach. Gordon et al. [15] proposed a framework to manage cyber-risk. The antecedent activities involve the assessment of the risk involved in a security breach. Subsequent steps involve the preventive measures necessary to avert such an attempt. These measures are divided into technical or procedural measures (i.e., access control, firewalls) and financial measures (such as buying cyber insurance). The final step entails the maintenance of accepted level of risk.

The majority of current research on information security focuses on the preventive measures required for reducing cyber-risk. There is a large body of research that describes the technical aspects of security [14] such as encryption and secure communications, access control, and intrusion detection. This research can help managers select the technical preventive measures that best fit their organizational needs. Similarly, research addressing the behavioral aspects of security breaches (e.g., [37]) can help managers understand procedural preventive measures. However, there is a relatively small but growing body of academic research that can help managers assess the economic threats and financial vulnerabilities caused by information security breaches (for examples see [11, 14, 20, 26]). The goal of this paper is to add to this body of knowledge by assessing the financial impact of virus attack announcements on attacked companies.

In the following section, we describe the reasons for choosing market value as a measurement of the economic impact of security breaches. Section 3 describes the characteristics of virus attacks and defines them as unexpected events. Section 4 introduces the financial measures of unexpected events. In Section 5, we detail the methodology used. In section 6, we introduce and analyze the study's results. In section 7, we discuss the results, the study's limitations, and future research.

2 Market Value

The economic impact of security breaches is of interest to companies trying to decide where to place their information security budget [15]. As the characteristics of security breaches change, companies continually reassess their IS environment for threats [23]. In the past, Chief Information Officers (CIOs) have relied on FUD – fear, uncertainty, and doubt – to promote IS security investments to upper management. Recently, some insurance companies have created actuarial tables that they believe provide ways to measure losses from computer interruptions and hacker attacks [34]. However, these estimates are questionable mostly due to the lack of historical data [15]. Some industry insiders confess that the rates for such plans are mostly set by guesswork [2]. As cited in Gordon et al., [15](p. 82): "These insurance products are so new, that the $64,000 question is: Are we charging the right premium for the exposure?" Industry experts cite the need for improved return on security investment (ROSI) studies that could be used by the organization to justify investments in security prevention strategies. However, assessing the financial loss from a potential IS security breach is a difficult step in the risk assessment process for the following reasons:

1. Many organizations are unable or unwilling to quantify their financial losses due to security breaches (for additional information see [32])
2. Lack of historical data. Many security breaches are unreported. Companies are reluctant to disclose these breaches due to management embarrassment, fear of future crimes [19], and fear of negative publicity [31]. Companies are also wary of competitors exploiting these attacks to gain competitive advantage [31].
3. Additionally, companies may be fearful of negative financial consequences resulting from public disclosure of a security breach [16].

Justifying investments in IS security using ROSI measures is difficult to accomplish. If the security measures work, the number of security incidents is low and there are no measurable returns. Accounting based measures such as ROSI are also limited by the lack of time and resources necessary to conduct an accurate assessment of financial loss when companies' IT resources are devoted to understanding the latest technologies and preventing future security threats [25]. In addition, potential intangible losses such as "loss of competitive advantage" that result from the breach and loss of reputation [8] are not included in ROSI measures because intangible costs are not directly measurable. Therefore, there is a need for a different approach to assess the economic impact of security breaches. One such approach is to measure the impact of a breach on the market value of a firm. A market value approach captures the capital market's expectations of losses resulting from the security breach. This approach is justifiable because often companies are impacted more by the public relations exposure than by the attack itself [16]. Moreover, managers aim to maximize a firm's market value by investing in projects that either increase shareholder value or minimize the loss of shareholder value. Therefore, in this study we elected to use market value as a measure of the economic impact of security breach (virus attack) announcements on companies. In the following section we define a security breach as an unexpected event and discuss the characteristics of virus attacks.

3 Virus Attacks and their Reported Impact

An IS security breach is a violation of an information system's security policy. While security has long been a concern for IS managers, reports of serious security breaches have become more frequent in today's networked environment. The explosion of the World Wide Web (WWW) and the subsequent growth of e-commerce increase the exposure of organizations to external security breaches. Evidence of the current state of Internet security can be found in a recent CSI/FBI Computer Crime and Security Survey [32]. In the last four years, Internet connectivity has been cited as the primary source of attacks (78%). The most commonly reported security breaches are virus attacks [32]. Virus attacks reportedly cause billions of dollars in damage and have been accelerating in their scope and severity. Thus, we selected to study the financial impact of virus attacks as an upper bound exemplar of security breaches.

A virus is a small piece of self-replicating computer code that attaches itself to a larger, legitimate program [27]. While acknowledging the potential existence of harmless or even productive viruses (as described in [7]), the discussion in this paper is limited to viruses that are created with the purpose of causing damage. Early viruses were static pieces of code that copied themselves from program to program or diskette to diskette [29]. These viruses were easily contained – causing limited damage. Today's viruses are significantly more complex, which makes detection and removal more difficult. The most common types of viruses include macro viruses, e-mail viruses, trojan horses, and worms. In our discussion we term them all viruses.

While the threat of viral attacks was evident in the early 1980s, the first widely seen viruses did not occur until later in the decade. By 1988, virus attacks against IBM PCs, Apple II computers, and Macintosh computers had been reported [17]. The emergence of computer networks and the Internet in particular has created a new means for spreading computer viruses. Robert Morris is responsible for the first known viral attack against the Internet [35], which infected nearly 6,200 individual machines (about 7.3% of the Internet's computers at the time) and caused 8 million hours of lost access and an estimated $98 million in losses [26]. Since the Robert Morris worm, the Internet has been the victim of numerous viral attacks (such as Jerusalem, Chernobyl, and Michelangelo). However, until the mid 1990's access to the Internet was limited by the "Acceptable Usage Agreement", thus limiting the potential impact of virus attacks. Only after the commercialization of the Internet in 1994 was the Internet available to the general public, leading to an increasing number of virus attacks that infected a large number of commercial organizations and caused accelerated financial damage. For example, in March 1999, the Melissa virus forced a number of large companies to shut down their e-mail systems, causing an estimated $80 million in damages [5]. In May 2000, the LoveLetter worm (i.e., the I Love You virus) caused an estimated $100 million in damage by infecting some 1.27 million computer files worldwide, with nearly 1 million in the United States [18]. In July 2001, the Code Red worm spread at an unprecedented rate, doubling its infestation rate every 37 minutes, eventually infesting over 350,000 hosts [28] and causing an estimated $2 billion in damage [30]. In January 2003, the Slammer worm infected about 90% of all vulnerable hosts on the Internet [28]. In August 2003, the Blaster worm affected nearly 500,000 computers in its first week [6]. ICSA labs estimated remediation costs (including hard, soft, and productivity costs) of $475,000 per company for the Blaster worm.

4 Financial Impact of Unexpected Event

Following the taxonomy of computer security incidents developed by Howard and Longstaff [21], a virus attack can be classified as a single computer and network security event involving an action directed against a specific target. In this case, the action is a virus attack and the target is a particular computer or a network of computers. Within the taxonomy, not all events are considered likely or even possible to occur. Therefore, we consider an Internet security breach (such as a virus attack) to be a negative computer security event that is not expected to occur on a regular basis. Prior research has assessed the financial impact of various unexpected events using both market-based measures and accounting-based measures of performance. However, the more popular research approach has been the event study. The event study examines the stock market reaction to the public announcement of a particular event and is based on the efficient market hypothesis [10]. According to the semi-strong form of the efficient market hypothesis, the market price of a firm fully reflects all publicly available information [12]. Therefore, an abnormal stock return associated with an unexpected event should be observed and measurable if the event has information content [22]. Previous research suggests that public news of an event that is generally seen as negative will cause a drop in a firm’s stock price (e.g., [1]). Sprecher and Pertl [36] found that firms experiencing a loss from a catastrophic event sustained an immediate adverse effect on their stock price. Overall, prior studies of negative, unexpected events indicate that the market penalizes announcing firms in the first few days following the public disclosure of the negative event. However, it is unclear if firms suffer similar penalties following an announcement of a virus attack.

Despite the impact of IS security breaches on organizations and the heavy financial impact reported in trade magazines, there have been very few academic studies on the topic. Ettredge and Richardson [11] assessed the market risk associated with electronic commerce (e-commerce) activity. They performed a study to measure the spillover effect in the stock market response to a series of Denial-of-Service (DOS) attacks against several of the best-known Websites in February 2000. Results showed that investors do perceive risk in e-commerce activities as the DOS attacks had a larger negative spillover market impact on Internet firms than on non-Internet firms. Hovav and D’Arcy [20] found that DOS attacks have little effect on the market value of attacked companies. However, these attacks have a larger impact on E-commerce companies whose core business depends on their Web presence than on non-Internet specific companies. McAfee and Haynes [26] conducted the only study to estimate the impact of virus attacks. They calculated the damage of the Robert Morris worm using accounting-based measures including direct programmer costs, indirect labor and burden costs, and indirect costs such as lost machine down time and user lost access time. Given the increase in the number of virus attacks over the last 15 years and the increase in their severity, it is imperative to evaluate the economic impact of these attacks. As described above, prior research found that public announcements that contain negative information cause an abnormal drop in the stock value of affected companies. Therefore, we anticipate that virus attack announcements will have a negative impact on the stock value of attacked companies.

H1: An announcement of a virus attack of a company *j* will result in negative abnormal returns on stock *j* for the day of the announcement.

Traditional event studies look at the distribution of the cumulative standardized abnormal returns (CSAR) of all affected companies. The virus attacks are expected to have a negative impact on the CSAR of the sample (i.e., the total of the actual returns << total expected returns).

H2: The cumulative standardized abnormal returns for the entire sample during the event period are significantly negative.

The following section depicts the methodology used. The data collection and analysis conform to the conventional procedures used in event studies.

5 Methodology

A procedure for sample selection similar to the method used by Subramani and Walden [38] and Im et al. [22] was followed in this study. We collected data on virus attacks using a search of business news in the Lexis-Nexis database. The search consisted of all public announcements of virus attacks between 1988 and 2002 resulting in 224 announcements. The initial list was then refined and evaluated based on the following criteria:

1. Only announcements by firms publicly traded on either the New York Stock Exchange (NYSE) or the NASDAQ stock exchange were included.
2. Announcements that might be confounded by other key firm notices such as mergers, acquisitions, earnings, stock splits, dividends, etc. within five days of the virus attack announcement were excluded.
3. To remove event day uncertainty [9], we triangulated our Lexis-Nexis search results with additional Web searches and information from financial publications.

For individual firms' stock market data, we relied on the database of the Center for Research in Security Prices (CRSP). We included in the sample only virus attack announcements for which stock return data was available. These sampling criteria yielded 186 virus attack announcements (events). The impact of announcements of virus attacks on common stock prices is computed using event study methods commonly employed in the accounting and finance literature [10]. The event of interest in this study is the public announcement of a virus attack by either the attacked firm or some other media outlet. If an announced virus attack contains new information, it should cause the markets to revalue the firm. Determining whether these events affect a firm's stock price requires that we estimate what the firm's stock price would have been had there been no announcement. We then calculate the standardized abnormal returns. Under the null hypothesis of zero expected abnormal returns, Z is approximately unit normally distributed (see [24]). For a more detailed discussion of analytical techniques employed in event studies, see Campbell et al. [4].

6 Analysis and Results

To test hypothesis 1, we calculated the mean abnormal return for each individual company, analyzed the results, and assessed the impact. Table 1 summarizes our findings. Overall, the results indicate that the virus announcements did not result in negative abnormal returns over any of the five event periods for our sample of attacked companies, as the mean abnormal return for each event period was positive. Thus, hypothesis 1 was not supported. However, there is partial support for hypothesis 1 as almost half of the firms experienced negative abnormal returns (Table 1) for a period of 25 days after the announcement.

Table 1. Mean Abnormal Returns and Number of Negative Returns for Attacked Companies

Event Windows	Mean Abnormal Return	Median Abnormal Return	Number of Negative Abnormal Returns
[0, 0]	0.0032	0.0019	79 (42%)
[0, 1]	0.0029	0.0010	81 (44%)
[0, 5]	0.0013	0.0016	79 (42%)
[0, 10]	0.0012	0.0013	82 (44%)
[0, 25]	0.0005	0.0007	84 (45%)

To test hypothesis 2, we calculated the CSAR for the entire sample. Table 2 lists the mean CSAR for each event window as well as the results of the z-tests to test the significance of the CSAR. Average CSARs for each of the event periods are positive, indicating that the virus attack announcements did not result in lower abnormal returns for the sample over any of the time periods. These results are contrary to what was expected, and therefore we reject hypothesis 2.

Table 2. Cumulative Standardized Abnormal Returns (CSAR) for Attacked Companies

Event Windows		Mean CSAR		Z-value*
[0, 0]		0.1196		1.6317
[0, 1]		0.0787		1.0730
[0, 5]		0.0554		0.7550
[0, 10]		0.0380		0.5183
[0, 25]		0.0134		0.1829

* Z- statistic to compute the significance of the average abnormal return over each event period under the null hypothesis that the average abnormal return is zero.

To further test hypothesis 2, we divided the virus announcements into industry subsamples by the SIC (Standard Industrial Classification) code of the attacked company. Similar results were found analyzing the sample by industry (i.e., there is no industry impact on the results of the analysis). These results are displayed in Appendix A.

7 Discussion

Overall, the above results did not demonstrate that there is a significant impact of virus attack announcements on the share price of the attacked companies. Mean abnormal returns were positive for each of the event periods studied. In addition, CSARs were not significantly negative (for the total sample or by industry) over any of the five event periods, whereas viruses were associated with negative stock returns for about 44 - 45% of the attacked companies. These unexpected findings are contradictory to the increasing financial impact reported by trade magazines and may be due to one of the following: (1) the market anticipates the virus attacks and incorporates the projected losses into the stock value of companies; or (2) there is little awareness in the general public as to the real damage caused by virus attacks, thus the market does not react to such announcements; or (3) the financial damage reported in trade magazines is inflated and the above market analysis reflects a more rational view of the actual damages.

Our findings demonstrate that the market does not penalize firms when they are exposed to virus attacks which results in little incentives for managers to demand improved security in current Information Systems (i.e., trustworthy computing) from IT vendors[1]. This also supports Blumenthal's [3] assertion that IT vendors take little action to increase information technology security due to lack of demand from their users. Thus, the assumption that market forces can be used as means to control security breaches and to increase the trustworthiness of computer systems might be false.

The above discussion suggests the need for further research in this area. First, there is a need to better understand the actual economic and financial impact of security breaches and their reflection on the market. Second, it is unclear if other types of attacks will have a more significant impact on shareholders' value. For example, recent legislation places legal liability on companies that expose private information to unauthorized entities (e.g., HIPAA, California's Database Breach Notification Security Act –SB 1386). Liability lawsuits may introduce new costs that could be perceived (by the market) as more substantial than the cost to recover from a virus attack. Therefore, it is possible that security breach announcements that involve the exposure of private information will result in more significant negative abnormal returns. Taxonomy of security breaches and the extent of their impact will allow managers to concentrate their efforts and allocate security budgets towards breaches

[1] For example, Microsoft's trustworthy computing initiative is estimated to cost $200 million and already delayed the launch of Server 2003 by several months. These additional costs will ultimately be transferred to the customer. Given that virus attacks do not reduce shareholder value, managers will have little incentive to demand increased security from IT vendors, which will only increase firms' IT costs.

that have larger effect. Third, there is a need to understand the impact of viruses on IT vendors and the factors that will drive the IT industry to create more secure information systems. In addition, future research can examine the impact of virus attacks on small and private organizations that may not have the resources to quickly recover from such attacks.

This study has several limitations. First, our sample contained two time clusters involving the Melissa virus in March 1999 and the LoveBug virus in May 2000. Time clusters can increase the significance of the results [9]. We repeated the analyses without the announcements involving these two virus events and the overall results of the study did not change. Second, the sample consists of only publicly traded companies. Therefore, the results cannot be generalized to non-publicly traded companies. Finally, many of the attacks caused a short downtime. Therefore, it is possible that the stock value was down during the day but closed normal once the problem was fixed and the affected systems were functioning again. This is referred to as intra-day stock movement.

8 Conclusions

Reports of security breaches in the popular business press suggest that computer viruses cause substantial financial damage to attacked companies. In this paper, we assessed the impact of virus announcements on attacked companies over a period of 15 years using event study methodology. Our results indicate that in general the market does not penalize companies who are victimized by virus attacks. These results are contradictory to findings in prior research, which indicates that the market penalizes companies involved in events containing negative information. These results also suggest that market forces cannot be used as a means of controlling security breaches nor can they be used to entice IT vendors to increase the trustworthiness of computer systems. Further research is required to understand the risk associated with security breaches. In addition, recent legislation suggests the need to better understand the factors that will reduce security risks and lead to a trustworthier Information Technology environment.

References

1. Baginski, S. P., R. B. Corbett, et al. (1991). "Catastrophic Events and Retroactive Liability Insurance: The Case of the MGM Grand Fire." The Journal of Risk and Insurance 58(2): 247-260.
2. Berinato, S. (2002). Finally, a Real Return on Security Spending. CIO: The Magazine for Information Executives. 15: 42-52.
3. Blumenthal, M. (1999). "The Politics and Policies of Enhancing Trustworthiness for Information Systems." Communication Law & Policy 4(4): 513-555.
4. Campbell, J. Y., A. W. Lo, et al. (1997). Event Study Analysis. Chapter 4 in The Econometrics of Financial Markets. Princeton, NJ, Princeton University Press.

5. Chen, C. Y. and G. Lindsay (2000). Viruses, Attacks, and Sabotage: It's a Computer Crime Wave. Fortune. 141: 484-487.
6. Chen, T.M., (2003). “Trends in Viruses and Worms.” The Internet Protocol Journal 6(3): 23-33.
7. Cohen, F (1984). Computer Viruses: Theory and Experiments. Proceedings of the Second IFIP International Conference on Computer Security, Toronto, Ontario, Canada.
8. D'Amico, A. (2000). What Does A Computer Security Breach Really Cost?, The Sans Institute. 2000.
9. Dyckman, T., D. Philbrick, et al. (1984). “A Comparison of Event Study Methodologies Using Daily Stock Returns: A Simulation Approach.” Journal of Accounting Research 22: 1-30.
10. Etebari, A., J. O. Horrigan, et al. (1987). "To Be or Not To Be - Reaction of Stock Returns to Sudden Deaths of Corporate Chief Executive Officers." Journal of Business Finance & Accounting 14(2): 255-279.
11. Ettredge, M. and V. J. Richardson (2001). Assessing the Risk in E-Commerce. Twenty-Second International Conference on Information Systems, New Orleans, LA.
12. Fama, E., L. Fisher, et al. (1969). "The Adjustment of Stock Prices to New Information." International Economic Review 10: 1-21.
13. Glover, S., S. Liddle, et al. (2001). Electronic Commerce: Security, Risk Management, and Control. Upper Saddle River, NJ, Prentice Hall.
14. Gordon, L.A. and M.P. Loeb (2002). “The Economics of Information Security Investment.” ACM Transactions on Information and Systems Security 5(4): 438-457.
15. Gordon, L.A., M.P. Loeb, et al. (2003) “A Framework for Using Insurance for Cyber-Risk Management.” Communications of the ACM 46(3): 81-85.
16. Hancock, B. (2002). “Security Crisis Management - The Basics.” Computers & Security 21(5): 397-401.
17. Hayes, F. (2003). The Story So Far. Computerworld. 37: 26-27.
18. Hinde, S. (2000). "Love Conquers All?" Computers & Security 19(5): 408-420.
19. Hoffer, J. A. and D. W. Straub (1989). "The 9 to 5 Underground: Are You Policing Computer Crimes?" Sloan Management Review (Summer 1989): 35-43.
20. Hovav, A. and J. D’Arcy (2003) “The Impact of Denial-of-Service Announcements on the Market Value of Firms.” Risk Management and Insurance Review, 6(2): 97-121.
21. Howard, J. D. and T. A. Longstaff (1998). A Common Language For Computer Security Incidents. Pittsburgh, PA, CERT Coordination Center at Carnegie Mellon University: 1-33.
22. Im, K. S., K. E. Dow, et al. (2001). “A Reexamination of IT Investment and the Market Value of the Firm: An Event Study Methodology.” Information Systems Research 12(1): 103-117.
23. Kelly, B. J. (1999). "Preserve, Protect, and Defend." Journal of Business Strategy (September/October 1999): 22-26.
24. Loderer, C. and D. C. Mauer (1992). “Corporate Dividends and Seasoned Equity Issues: An Empirical Investigation.” Journal of Finance 47(1): 201-225.
25. Lyman, J. (2002). In Search of the World's Costliest Computer Virus, www.newsfactor.com/perl/story/16407.html. 2002.

26. McAfee, J. and C. Haynes (1989). Computer Viruses, Worms, Data Diddlers, Killer Programs, & Other Threats To Your System. New York, New York, St. Martins Press.
27. Montana, J. C. (2000). "Viruses and the Law: Why the Law is Ineffective." The Information Management Journal 34(4): 57-60.
28. Moore, D., G.M. Voelker, et al. (2001). “Inferring Internet Denial-of-Service Activity.” Proceedings of the 10th USENIX Security Symposium, Washington, D.C.
29. Nachenberg, C. (1997). "Computer Virus - Antivirus Coevolution." Communications of the ACM 40(1): 46-51.
30. Panko. R.R. (2003). “Slammer: The First Blitz Worm.” Communications of the Association for Information Systems. 11: 207-218.
31. Power R. (2001). “2001 CSI/FBI Computer Crime and Security Survey.” Computer Security Issues and Trends 7(1): 1-18.
32. Power, R. (2003). “2003 CSI/FBI Computer Crime and Security Survey.” Computer Security Issues and Trends 9(1): 1-20.
33. Salierno, D. (2001). Managers Fail to Address E-risk. The Internal Auditor. April 2001: 13.
34. Salkever, A. (2000). Who Pays When Business Is Hacked?, www.businessweek.com/bwdaily/dnflash/may2000/nf00523d.htm.
35. Spafford, E. (1999). "Crisis and Aftermath." Communications of the ACM 32(6): 678-687.
36. Sprecher, R. and M. Pertl (1988). “Intra-Industry Effects of the MGM Grand Fire.” Quarterly Journal of Business and Economics 27: 96-16.
37. Straub, D.W. and R.J. Welke. (1998) “Coping With Systems Risk: Security Planning Models for Management Decision Making.” MIS Quarterly 22(4): 441-469.
38. Subramani, M. and E. Walden (2001). “The Impact of E-Commerce Announcements on the Market Value of Firms.” Information Systems Research 12(2): 135-154.

Appendix A. Cumulative Standardized Abnormal Returns (CSAR) for Attacked Companies by Industry

Event Windows		Mean CSAR		Z-value*
Finance, Insurance, and Real Estate (n=25)				
[0, 0]		0.0919		0.4596
[0, 1]		0.1260		0.6300
[0, 5]		0.1309		0.6546
[0, 10]		0.0061		0.0303
[0, 25]		-0.0261		-0.1305
Manufacturing(n=78)				
[0, 0]		0.0911		0.8047
[0, 1]		0.0835		0.7374
[0, 5]		0.0242		0.2136
[0, 10]		0.0233		0.2062
[0, 25]		0.0100		0.0883
Retail Trade (n=6)				
[0, 0]		0.2835		0.6944
[0, 1]		-0.1170		-0.2866
[0, 5]		0.0440		0.1077
[0, 10]		0.0412		0.1009
[0, 25]		0.0436		0.1067
Services (n=35)				
[0, 0]		0.1462		0.8649
[0, 1]		0.0692		0.4094
[0, 5]		-0.0393		-0.2325
[0, 10]		-0.0116		-0.0687
[0, 25]		-0.0139		-0.0821
Transportation, Communications, Electric, Gas and Sanitary Services (n=42)				
[0, 0]		0.1436		0.9306
[0, 1]		0.0774		0.5017
[0, 5]		0.1488		0.9641
[0, 10]		0.1251		0.8110
[0, 25]		0.0617		0.3999

* Z statistic to compute the significance of the average abnormal return over each event period under the null hypothesis that the average abnormal return is zero.

A Formal Security Model for Collaboration in Multi-agency Networks

Salem Aljareh
Computing Science, Newcastle University, Newcastle upon Tyne, UK NE1 7RU
Email: s.s.aljareh@ncl.ac.uk

Nick Rossiter (Informatics), Michael Heather (Law)
University of Northumbria, Newcastle upon Tyne , UK, NE1 8ST
Email:nick.rossiter@unn.ac.uk; m.heather@unn.ac.uk

Abstract: Security problems in collaborative work between multiple agencies are less well understood than those in the business and defence worlds. We develop a perspective for policies and models that is task-based on a need-to-know basis. These policies are represented by two protocols, the first CTCP (Collaboration Task-based Creation Protocol) dealing with negotiation, decision and agreement between the parties involved and the second CTRP (Collaboration Task-based Run-time Protocol) responsible for the operation of the policy. The two protocols and the relationship between them are defined in Petri-Nets. The overall model is formally defined using a categorical pullback construction. Each of the protocols, represented as Petri-Nets for state-transition purposes, is a category-valued functor in the pullback.

1 INTRODUCTION

Information is naturally sharable among groups such as team, committee, organization, country and federation in a manner based on trust. However to achieve an accepted level of trust is quite a complicated issue because as the collaboration grows wider, more participants are involved with divergent policies. Although designing secure models for collaboration environments has been a target of a number of academic and commercial research bodies and several works have been done (both theoretical and practical), numerous organizations still keep their systems (especially the trusted systems) unconnected with outsiders.

Basically security systems are built out of the available mechanisms to meet a security policy based on a selected security model [Gollmann, 1999]. A review has been made elsewhere [Aljareh & Rossiter, 2002] of the appropriateness of standard security models for collaborative multi-agency systems. Most are either targeted at a specific security requirement or are too static to represent a dynamic situation. All deal with a single policy, whereas by definition the multi-agency and collaboration environment involves more than one policy.

A motivating example of an application that involves multi-agency services is the medical information services. The only model designed to meet the security requirements for the medical records in the UK was the BMA (British Medical Association) Security Policy Model [Anderson, 1996]. This model was recently examined [Aljareh & Rossiter, 2001] against the multi-agency security requirements and it was found that the issue of sharing clinical information including collaboration activities with other agencies such as police, social services or the education authority was not clearly considered. For instance the *need-to-know* problem was not addressed in the BMA model, as the BMA does not accept that *need-to-know* is an acceptable basis for access control decisions. However there might be a case where *need-to-know* cannot be avoided. For instance a service provider such as an insurance company offers its services conditioned by some information about the patient who applies for such services. An example is given in [Aljareh & Rossiter, 2001].

In this paper, we propose a security model that we argue will alleviate the security difficulties that may arise in attempts to build a collaboration network. The model is constructed from a task-based perspective, as this approach seems to offer the best way forward, as discussed later. An example of a prototype for informal collaboration, handled using the model, is given elsewhere [Aljareh & Rossiter, 2002]. The general principles of the model are discussed and a diagrammatic notation is devised. Two task-based collaboration protocols, expressed in this paper in the form of Petri-Nets, represent the permitted states and transitions. The choice of Petri-Nets as the notation is discussed. Finally the overall model is constructed formally as categorical pullbacks to illustrate its foundation on established logical principles.

2 A TASK-BASED PERSPECTIVE FOR COLLABORATION NETWORKS

A collaboration business, by definition, is based on the needs of the collaborators from each other. Each side needs information or a service from the other participants. The obvious question that someone will immediately ask before he/she releases any confidential information or responds to an enquiry is: what for? For what purpose is the information required? Usually the expected answer will be the naming of a task for which the information required is essential, sometimes with a further explanation of the benefit of this task for the two sides (collaboration proposal). The information owner may like to restrict the use of this information by some conditions (security policy). If they reach initial agreement a detailed negotiation will then take place until they reach a considered level of trust, which leads to a collaboration agreement to perform the task. One reasonable condition might be to limit the use of the information by other tasks. For instance it could be specified that the information should not be used outside the task for any purpose.

We have decided to construct our model as a task-oriented model for the following reasons:

1. Fundamentally any collaboration scheme is based on specific tasks: there is no collaboration without a task.
2. The task-based approach is promising to address the need-to-know problem, satisfying a user requirement in any multi-agency services environment.
3. The collaboration task is the common object between the collaborators.
4. Shared information ownership can be granted to the collaboration task.
5. The task is scalable, flexible and dynamic.
6. Explicit responsibility is recognized in the task-based approach.

Overall the basis for any collaboration is an aim to share resources in order to achieve common benefits by performing shared operations. Other task-based approaches to security are discussed later.

3 GENERAL PRINCIPLES FOR OUR MODEL

Collaboration: In our model we consider any deal/trade between individuals or groups, which aims to benefit the sides involved as a kind of collaboration. The following are some forms of collaboration:

- Trading between customers and service providers.
- Joint operation projects
- Research group collaboration.
- The clinician and the patient trade/relationship: the clinician's job exists because of the patient, and the patient needs the clinician for treatment. So both need each other and benefit each other. The clinician may need to know some information from the patient as part of the course of treatment. The relationship is in general based on trust. In this example there are two sides trading benefits through the task called treatment

Ownership: In this model an item of information is owned initially by its natural owner that is the person to whom the information relates. For instance information about the baby is owned by the baby although this information is controlled by guardian/parents. In computer security terms this is called *grant access* or *delegation.* Once this information is required to be shared among collaboration parties, an access will be granted to what we call the *collaboration-task*, controlled by the *task-policy*. The information owner and/or the access controller will be part of the negotiation that results in the task policy.

Authorization: A participant in a collaboration network, called task-participant, will be authorised to gain access to a collaboration-task. This authority will be limited by what we call task-policy.

Responsibilities: All responsibilities should be explicitly defined in the task policy. This way each individual collaborator (task-participant) knows their responsibilities such as the required duties, the rules to follow (including ethical codes), the limitations (e.g. time, use of material and information) and the penalties.

4 COLLABORATIVE TASK CHARACTERISTICS

The characteristics of collaborative tasks are considered to be:

1. Flexible: can be a single activity or group of activities sharing same policy, each of which can be selected as the need arises.
2. Dynamic: can be updated even while it is running (supporting post-hoc justification). For instance a nurse can be replaced by another one if he/she is not, for any reason, able to complete his/her duty in a surgical operation. However any change in the task elements should be fully and carefully documented.
3. Secure: should be fully protected using all the available mechanisms.
4. Scalable: can be upgraded, for instance to fill some gaps in the original task. A new collaboration task can be built starting from default tasks.

5. Accountable: all collaboration protocol states and all task run-time events of the collaboration must be well documented.

5 DIAGRAMMATIC REPRESENTATION OF MODEL

The architecture in Figure 1 illustrates the general components of our model. The main component is the collaborators (two or more), each of which will need to define three elements: requirements (what does he/she/it/they aim to gain from the other side),

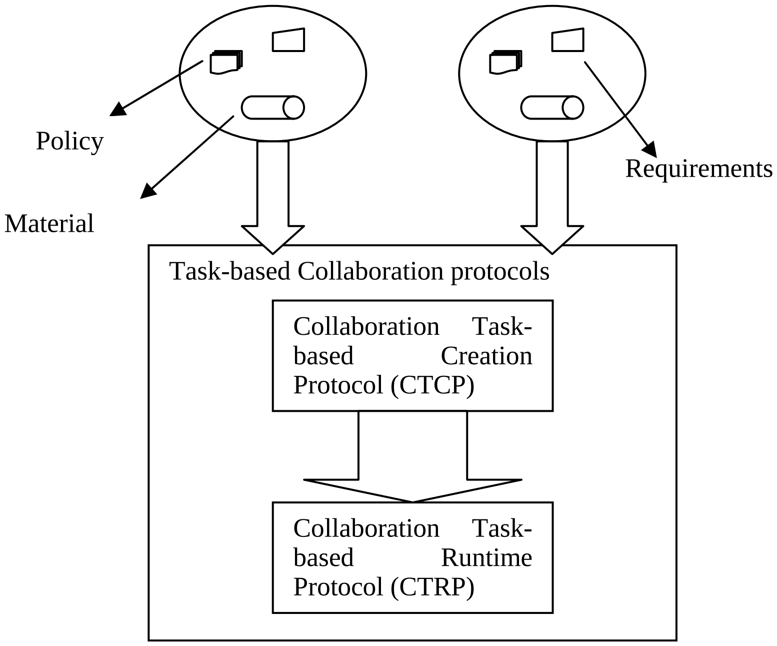

Policy
Material
Requirements
Task-based Collaboration protocols
Collaboration Task-based Creation Protocol (CTCP)
Collaboration Task-based Runtime Protocol (CTRP)

Figure 1: General Architecture for secure Collaboration Environment

policy (rules that need to be obeyed) and material (e.g. information to release or services to offer). The second component is a pair of task-based collaboration protocols -- the Collaboration Task Creation Protocol (CTCP) and the Collaboration Task Runtime Protocol (CTRP) -- both detailed later in the following sections.

CTCP includes a negotiation between all collaborators where the proposed task will be discussed including all collaborators' policies and requirements. This process (negotiation) continues until a decision is taken either by rejecting the proposal or by accepting it. The acceptance of a proposal will lead to a formal agreement/contract, which will produce the proposed collaboration task in its final stage including all of the policies and requirements. Negotiation can of course be a very complex task [Chu-Carroll & Carberry, 2000]. The work described here could be extended later to include such aspects as conflict resolution. CTRP will start after a successful compilation of CTCP and as scheduled .in the *task_policy* (not necessarily immediately after the end of CTCP).

The main function of CTRP is to process the task that was previously created by the CTCP protocol and ensure that the *task_policy* is obeyed, the collaborators are aware of the circumstances and the right action is taken.

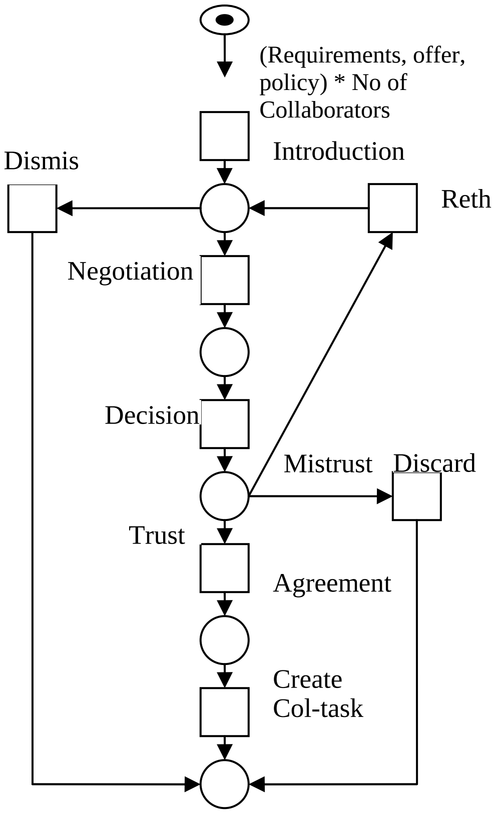


Figure 2: Petri-Net Graph representing the Collaboration Task-based Creation Protocol (CTCP)

6 REPRESENTATION OF PROTOCOLS IN PETRI-NET NOTATION

We use the Petri-Nets model to represent our collaboration protocols to provide a formal basis and a more applicable medium for computer scientists. Flow charts lack a formal basis and can be ambiguous in representing states and transitions. Data flow diagrams emphasise flows of data, not states, which are considered critical in security systems.

Net theory was originally introduced in a PhD thesis of C. A. Petri. Later Reisig [1985] introduced it to the software engineering area. More recent advances in this formalism are described in [Reisig & Rozenberg, 1998]. The usefulness of Petri-Nets in providing a theoretical basis for handling object life cycles has been demonstrated by van der Aalst and Basten [2001]. In collaboration networks, similar to the multi-agency services investigated here, Furuta and Stotts [1994] presented an evolution of the Trellis model by providing a formal Petri net basis for prototyping the control of such a network.

In the security area an industrial use of Coloured Petri-Nets was developed by Rasmussen and Singh, [1996] making it possible to perform simulations. The nets were debugged by constructing reachability graphs. Joshi and Ghafoor [2000] specified a multi-level security model for multimedia using a time- augmented coloured Petri-Net model. For cryptographic protocols Crazzolara and Winskel [2001] use Petri-Nets to illustrate how their semantics can be used to prove security properties. Ryan [2003] notes that causality, critical in the analysis of security protocols, is closely related to information flow and that causal structures are rather more explicit in Petri-Nets than in many other areas.

In general Petri-Nets have been widely used for the modelling and analysis of systems that are characterized as being concurrent, asynchronous, distributed, parallel and non-deterministic [Jensen, 1996]. All these features apply in the collaborative, multi-agency systems studied here. Activities in the systems: a) overlap in timing; b) are run independently rather than according to some common time signal; c) are run over many different servers; d) involve the splitting of tasks into subtasks which run in parallel until some common join point is reached; and e) may not give the same result in negotiation each time the protocols are run.

The Petri-Net in Figure 2 represents the CTCP protocol. The initial state represents for each collaborator their requirements, policies and offers. For instance, in the patient-doctor collaboration, the patient's requirements are treatments, the patient's policy is to keep personal information secret, the doctor's requirements may include information about the patient and the doctor's offer is a treatment course. Following discussion of this initial state the task, at first an offer from one side or a requirement from another, is accepted as an offer for further negotiation or rejected without any further details. Policy considerations are normally omitted during the introduction transition.

If the proposed task is found to be reasonable then all collaborators will enter into a detailed negotiation in which all aspects including requirements, services and polices will be clarified for all collaborators. After that one of three decisions will be taken: the first option could be one of the collaborators needs more time to think about the task/offer; the second option could be that the expected level of trust could not be ensured so the task is simply dismissed; the third option is that all collaborators trust each others so that an agreement between all collaborators will take place. This agreement at the end will be formulated in what we call the collaboration task.

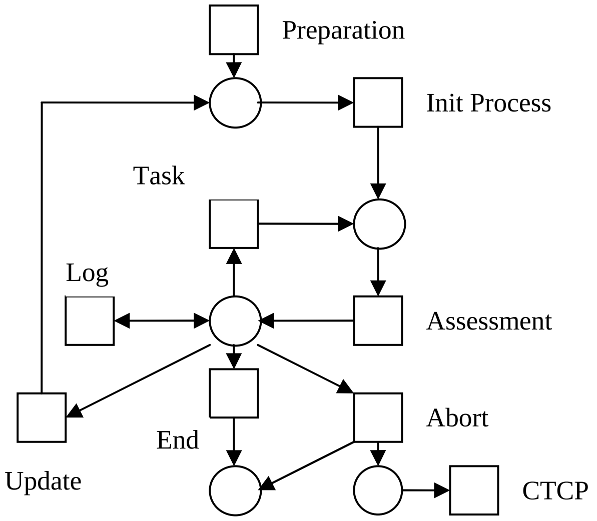


Figure 3: Petri-Net Graph representing the Collaboration Task-based Run-time Protocol (CTRP)

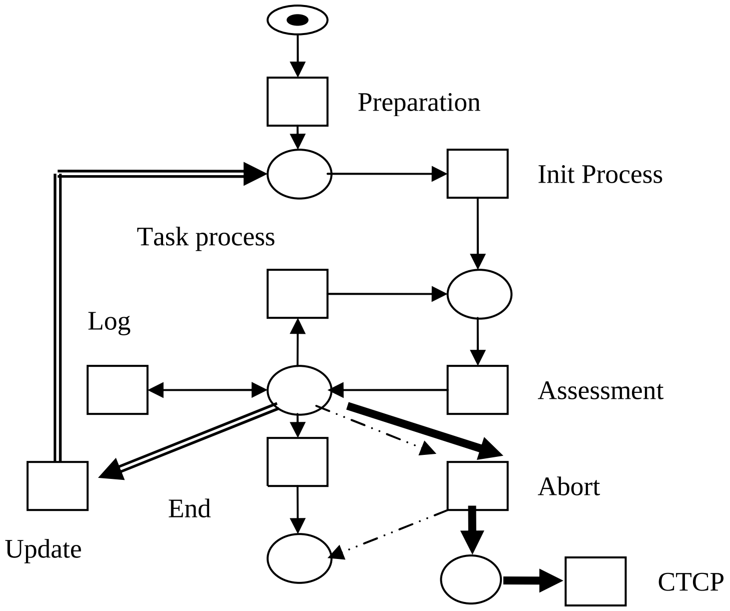


Figure 4: Exception occurring during CTRP, followed by an abort and a return to CTCP (see section 6 for notation)

This task will be limited in scope by the task policy, which is a composition of all collaborators' policies, meeting all sides’ requirements.

The Collaboration Task Runtime protocol (CTRP), illustrated in Figure 3, starts after the task has been completely created by the CTCP protocol and when its schedule time, according to the task-policy, is due. Before starting the process of the task some tasks need some preparations. Then the task process starts following the policy that has been approved in the CTCP stage. Each state of this process is monitored, assessed (verified against the task-policy) and then documented. The task assessment may result in one of the following:

1. The task is proceeding satisfactorily, following the policy and the plan and has not finished yet, so the task should persist.

2. The task needs an update to meet its requirements. Depending on how the updates affect the process: the task may restart or continue from the last state of the process.

3. The task reaches its scheduled end, hence the task terminates normally.

4. There might be a case where the task abnormally terminates, for instance the task-policy has been violated, or the task exceeds the scheduled time without valid reasons. The abnormal termination could lead either to the end of the task and then the collaboration or to a new session of the CTCP. An exception is raised when the policy has been violated as in Figure 4.

In our model exceptions are divided into three types according to the handling process:

1. Exceptions with which the task can still continue to its normal end. Exceptions of this type are handled within the CTRP protocol by the task update component. Figure 4 shows the path of the exception type as a double line =..
2. Exceptions with which the task must be terminated and another task is required to complete the planned function. Such cases are handled partially in the CTRP protocol. The task in such cases is aborted and the process log (task history) used by the CTCP protocol to create another task to redo the function that could not be done by the terminated task in view of the exceptions that have arisen. The exception handling path for this type is shown as a thick line in Figure 4.
3. Exceptions with which the task must be terminated and there is no need for any further actions. There are cases where the task immediately terminated and no further actions are possible. Exceptions from this type are handled within the CTRP protocol through the ABORT component. The exception handling path for this type is shown as a dotted line in Figure 4.

7 FORMALISATION WITH CATEGORICAL PULLBACKS

The relationship between the protocols CTCP and CTRP can be represented rigorously by the categorical pullback shown in Figure 5. Pullbacks are examples of cartesian closed categories [Mac Lane, 1998].

$$\begin{tikzcd}
 & \mathbf{C} \arrow[ld, shift right, "\eta_c"'] \arrow[rd, "\iota_c"] & \\
\mathbf{C\ X_B\ A} \arrow[ru, shift right, "\pi_{c\ x\ a}"'] \arrow[rr, shift left=2, "\exists"] \arrow[rr, shift right=2, "\forall"'] \arrow[rd, "\varepsilon_{c\ x\ a}"'] & & \mathbf{C/B} \arrow[ll, "\Delta"'] \\
 & \mathbf{A} \arrow[ru, "\iota_a"'] &
\end{tikzcd}$$

Figure 5: Categorical Pullback of System (A) over Environment (C) in the context of Purpose/View (C/B)

This figure shows the relationship between four categories (denoted in bold font). **C** is the complete environment, **A** is a particular system to which a user may require access, **C/B** is a slice category or subcategory of **C** and **C X_B A** is a limit, representing the relationship between **C** and **A** in the context of **B**. The limit can be viewed as a subcategory of the product **C X A** over **B**. Three functors map between **C X_B A** and **C/B.** ∃ is the existential quantifier selecting some **C/B** for a particular **C X_B A,** ∀ is the universal quantifier selecting **C/B** that satisfy all the rules determined by Δ as the diagonal functor selecting a limit **C X_B A** for a particular subcategory **C/B**. Δ is right adjoint to ∃ and left adjoint to ∀, written ∃ -| Δ -| ∀. Two natural transformations are shown. η_c is the unit of adjunction comparing objects C with objects C X_B A and ε_{cxa} is the counit of adjunction comparing objects C X_B A with objects A. η_c is an inverse projection (π*) and $\varepsilon_{c\ x\ a}$ is a projection (π).

In terms of our CTCP/CTRP model given above:

The diagonal functor Δ corresponds to the protocol CTCP whereby a limit **C X_B A** is selected for a particular purpose **C/B** through negotiation. CTCP selects a relationship between C and A for a particular purpose such that the diagram in Figure 5 commutes, that is ι_c o $\pi_{c\ x\ a} = \iota_a$ o $\varepsilon_{c\ x\ a}$. As a Petri-Net CTCP can be represented as a monoidal category [Asperti, Ferrari & Gorrieri, 1990]. CTCP is therefore a category-valued functor. **C X_B A** corresponds to the policy rules derived through the negotiation in CTCP.

The existential functor ∃ is a type constraint: there must exist for all policy rules in **C X_B A** an entry in the system **C/B**.

The universal quantifier functor ∀ corresponds to the protocol CTRP: all the rules held in the negotiated policy (the limit **C X_B A**) are applied for a particular purpose (**C/B**). Like CTCP, CTRP is a category-valued functor with its Petri-Net defined as a monoidal category.

Overall CTCP is right-adjoint to ∃ and left-adjoint to CTRP. CTRP is right-adjoint to CTCP.

Exceptions are much less likely to occur in the strongly typed categorical model than in a set model. If they did occur they would be handled at the natural transformation level. The unit of adjunction η_c is given as 1_c → CTRP o CTCP(c) and the counit of adjunction as CTCP o CTRP(c x a) → $1_{c\ x\ a}$. The former measures the change in c as the functors CTCP and CTRP are applied in turn. The latter measures the change in (c x a) as the functors CTRP and CTCP are applied in turn. The unit and counit both give a measure of consistency as the application is run with the possibility of exceptions being raised if divergence is noted.

8 DISCUSSION

We consider two aspects of our work. Firstly the extent to which task-based approaches have been used before in security systems; secondly the prospects for formal approaches in the security area.

The idea of task-based has been introduced before in a number of models [Fischer-Hübner & Ott, 1998; Steinke, 1997; Thomas & Sandhu, 1994]. All were at the basic level of this approach. The focus by the last two (Steinke; Thomas & Sandhu) was on whether a task-based security model could be an alternative authorisation and access control model to the subject-object traditional authorisation models. The first paper (Fischer-Hübner & Ott) tried to address the privacy problem using the task-based approach. We have intended in our model to use all of the power of this idea (task-based approach) to address the security problem of the collaboration networks and the multi-agency services environment. In more detail:

1. Steinke [1997] outlines the general features and characteristics of the task-based approaches such as:

- The need-to-know is related to the operation, which needs to be performed.
- Any information needs can be related to a task.
- Tasks are common entities that exist and relate directly to both users and to information.
- Tasks limit the access to the information from the start to the termination of the tasks.
- Tasks already exist, and are identifiable, flexible and dynamic.

The Group Security model (GSM) by Steinke was described as a security model, which provides access to information on the base of a user's task.

However some features of GSM are already rather obvious in existing information systems infrastructure. For instance in any relational database, it is always possible to grant users/roles to functions, procedures, and packages rather than grant them to the information objects (e.g. tables, views). These functions, procedures and packages are in fact tasks and group of tasks and also can be functionally minimized. GSM considers the discretionary security approach to deal with ownership. Overall GSM is more suitable for hierarchical systems, where the responsibilities are visible.

2. Thomas & Sandhu [1994] introduced the task-based approach initially as an approach to address integrity issues in computerized information systems from an enterprise perspective. Subsequently Thomas & Sandhu [1997] developed their approach to produce a paradigm for access control and authorisation management. The developed model is called Task-based authorisation control (TBAC).

3. Fischer-Hübner & Ott [1998] in their model attempted to address the privacy aspect using the task-based approach. The nature of the task-based approach eases the handling of the main privacy requirements such as:

- Purpose binding: personal data obtained for one purpose should not be used for another purpose without informed consent.

· Necessity of data collection and processing: the collection and processing of personal data shall only be allowed, if it is necessary for tasks falling within the responsibility of the data processing agency.

In contrast to the models of Steinke and of Thomas & Sandhu, this model takes a forward step to de-centralise the authorisation using a 4-eyes principle. However there were no end-user requirements supporting this model and the 4-eyes principle is not enough to ensure de-centralisation. The set theory which was used to represent this model is not proven, nor is it in a framework (Petri-Nets, Category theory, LaSCO, Ponder, VDM, Z, ...) where proof is done by following constructive principles or through a guaranteed mechanism. Finally the Fischer-Hübner & Ott model does not include collaboration ventures.

4. Mahling, Coury & Croft [1990] tried to build a task-based collaboration model. However this work starts from a relatively late stage in the negotiation where the plan, agreement and tasks are relatively clear. In addition their work does not consider the case of the multi-agency environments where the policies of the collaborators are different.

We argue that the real challenge for the task-based approach is the multi-agency services environment, where responsibilities are distributed and the ownership is dynamic. None of the existing approaches have considered the multi-agency aspects in detail.

Formalising security models is an important matter as this a way in which guarantees can be secured about the reliability of a model. Security rules for multi-agency systems need to be formulated at the policy level. At this level, category theory seems to be appropriate as it provides not only appropriate abstractions for this level but also, in a multi-level architecture, mappings to lower levels such as mechanisms. For interoperability a multi-level approach constructed in category theory has already proved very promising [Rossiter, Nelson & Heather, 2003]. The use of Petri-Nets, for detailed state transitions, within a categorical framework, for control of types and levels, looks to be a way forward for formalising security in information systems. More advanced techniques such as Timed Petri-Nets and Stochastic Petri-Nets should be useful in gaining greater expressibility. Validation techniques in Petri-Nets could also be used for verifying the model. The benefits of using Petri-Nets will be highest where collaboration occurs between multiple agencies. This is a natural area for applying Petri Nets with its concurrency, asynchronicity, distribution, parallelism and non-determinism.

9 CONCLUSIONS

This paper has introduced a task-based model to facilitate collaboration in trusted multi-agency networks. Our model is based on the fundamental aspect of the collaboration environment, which is the task-based perspective. Two task-based collaboration protocols (CTCP and CTRP), expressed in this paper in the form of Petri-Nets, are used to represent the permitted states and transitions. The extent to which task-based approaches have been used before in security systems has also been discussed.

The two protocols and the relationship between them are defined in Petri-Nets. The overall model is formally defined using a categorical pullback construction. Each of the protocols, represented as Petri-Nets for state-transition purposes, is a category-valued functor in the pullback. The use of Petri-Nets within a categorical framework looks to be a promising way forward for security problems.

REFERENCES

Aljareh, S., & Rossiter N., 2001, Toward security in multi-agency clinical information services, *Proceedings Workshop on Dependability in Healthcare Informatics*, Edinburgh, 22nd-23rd March 2001, 33-41.

Aljareh, S., & Rossiter, N., 2002, A Task-based Security Model to facilitate Collaboration in Trusted Multi-agency Networks, *ACM Symposium on Applied Computing (SAC) 2002*, Madrid, 744-749.

Anderson, R., 1996, A Security Policy Model for clinical Information Systems, *Proc. IEEE Symposium on Research in Security and Privacy,* 30–43.

Asperti, A., Ferrari, G. L., & Gorrieri, R., 1990, Implicative formulae in the `Proofs as Computations' analogy, Proc 17th ACM SIGPLAN-SIGACT Symp Principles Programming Languages, 59-71.

Chu-Carroll, J., and Carberry, S., 2000, Conflict Resolution in Collaborative Planning Dialogues, *International Journal of Human-Computer Studies,* 53(6) 969-1015.

Crazzolara, F., & G. Winskel, G., 2001, Petri-Nets in cryptographic protocols, *Proc. 6th International Workshop on Formal methods for Parallel Programming: Theory and Practice*, San Francisco

Fischer-Hübner, S., & Ott, A., 1998, From a Formal Privacy Model to its Implementation, *Proc. 21st National Information Systems Security Conference*, Arlington, VA.

Furuta, R, & Stotts, P D, 1994, Interpreted collaboration protocols and their use in groupware prototyping, Proceedings of the 1994 ACM conference on Computer supported cooperative work, Chapel Hill, North Carolina, United States, 121 – 131.

Gollmann, D., 1999, *Computer Security*. ISBN: 0 471 97844 2, John Wiley and Sons.

Jensen, K., 1996, Colored Petri-Nets - Basic concepts, analysis methods and practical use, Springer, second edition **1**.

Joshi, J., & Ghafoor, A., 2000, A Petri-Net Based Multilevel Security Specification Model for Multimedia Documents, *ICME2000, IEEE International Conference on Multimedia and Expo*, MP10.12 533, Purdue University, USA.

Mac Lane, S, 1998, *Categories for the Working Mathematician*, 2nd ed, Springer-Verlag, New York.

Mahling, D.E., Coury, B. G., & Croft, W. B., 1990, User Models in Cooperative Task-oriented environment. *Proc. 23rd Annual Hawaii IEEE International Conference on System Science*, 94-99.

Rasmussen, J. L., & Singh, M., 1996, Designing a Security System by Means of Coloured Petri-Nets. Proc. 17th International Conference in Application and Theory of Petri-Nets (ICATPN'96), Osaka, Japan, *Lecture Notes in Computer Science*, **1091** 400-419.

Reisig, W., 1985, *Petri-Nets: an Introduction*. Berlin; New York: Springer-Verlag.

Reisig, W., & Rozenberg G., 1998, Lectures on Petri-Nets: Advances in Petri-Nets. *Lecture Notes in Computer Science*, no. 1491.

Rossiter, N., Nelson, D. A., & Heather, M. A., 2003, Formalizing Types with Ultimate Closure for Middleware Tools in Information Systems Engineering, *5th International Conference on Enterprise Information Systems (ICEIS),* Angers, France 366-373.

Ryan, P, 2003, Theoretical Challenges Raised by Information Security, Workshop on Issues in Security and Petri-Nets (WISP), ICATPN.

Steinke, G., 1997, A Task-based Approach to Implementing Computer Security, *Journal of Computer Information Systems*, 47-54.

Thomas, R. K., & Sandhu, R. S., 1994, Conceptual Foundation for a Model of Task-Based Authorization, *Proc. 7th IEEE Computer Security Foundations Workshop*, Franconia, NH, 66-79.

Thomas, R. K., & Sandhu, R. S., 1997, Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management. *Proc. IFIP WG11.3 Workshop on Database Security*, Lake Tahoe, California pp.

Van der Aalst, W. M. P., & Basten, D., 2001, Identifying Commonalities and differences in Object Life Cycles using Behavioral Inheritance, Application and Theory of Petri-Nets 2001, *22nd International Conference ICATPN*, Newcastle, 32-52.

Security Analysis of MOR using $GL(2, R) \times_\theta \mathbb{Z}_n$

Christian Tobias

Justus Liebig University Giessen, Department of Mathematics
Arndtstrasse 2, 35392, Germany
Christian.Tobias@math.uni-giessen.de

Abstract. This paper cryptanalyses the MOR cryptosystem [6] when the group $GL(2, R) \times_\theta \mathbb{Z}_n$ proposed in [7] is used.
We show generic attacks on the system that work with every ring R. For a concrete choice of R even stronger attacks may be possible.

Key words: MOR cryptosystem, cryptanalysis, conjugacy problem

1 Introduction

In 2001 Paeng, Ha, Kim, Chee and Park proposed a new cryptosystem based on the difficulty of the discrete logarithm problem in the inner automorphism group $Inn(G)$ of a non-abelian group G [6]. Later this system was named MOR cryptosystem [7].

The used non-abelian group G has to be chosen very carefully not to undermine the security of the system. The first proposal for G was the semi-direct product group $SL(2, \mathbb{Z}_p) \times_\theta \mathbb{Z}_p$ (see [6]). The authors themselves showed the interrelation between MOR using $SL(2, \mathbb{Z}_p) \times_\theta \mathbb{Z}_p$ and MOR using $SL(2, \mathbb{Z}_p)$. Since the conjugacy and the special conjugacy problem can be efficiently solved in $SL(2, \mathbb{Z}_p)$, the security of MOR using $SL(2, \mathbb{Z}_p) \times_\theta \mathbb{Z}_p$ could be reduced to the hardness of the discrete logarithm problem in $SL(2, \mathbb{Z}_p)$ (see [7]).

In 2003 a detailed analysis of MOR using $SL(2, \mathbb{Z}_p) \times_\theta \mathbb{Z}_p$ [8] was published. The efficient modes of MOR using $SL(2, \mathbb{Z}_p) \times_\theta \mathbb{Z}_p$ proved to be extremely vulnerable to the presented attacks. In some cases an attacker is able to gain information equivalent to the secret key.

In [7] Paeng, Kwon, Ha and Kim described how to construct a semi-direct product group $GL(2, R) \times_\theta \mathbb{Z}_n$ from a given ring isomorphism $\Phi : R \to R$ and proposed to use this group for the MOR cryptosystem. The purpose of this article is to evaluate the level of security provided by MOR using $GL(2, R) \times_\theta \mathbb{Z}_n$. Our analysis focusses on the impact of the hardness of the computational Diffie-Hellman and the discrete logarithm problem in $< \Phi >$ on the security of MOR using $GL(2, R) \times_\theta \mathbb{Z}_n$. We show that if the computational Diffie-Hellman problem can be solved efficiently in $< \Phi >$, then the efficient modes of MOR using $GL(2, R) \times_\theta \mathbb{Z}_n$ are vulnerable to chosen-ciphertext attacks. Furthermore, if even the discrete logarithm problem can be solved efficiently in $< \Phi >$, then the secret key can be (partly) calculated from the public parameters.

The rest of this paper is organized as follows. In section 2 needed notations and definitions are described and the MOR cryptosystem is introduced. Section 3 shows how to construct a semi-direct product group $GL(2,R) \times_\theta \mathbb{Z}_n$ given a ring isomorphism $\Phi : R \to R$ and how to apply this group to the MOR cryptosystem. We further demonstrate that the discrete logarithm problem in $Inn(GL(2,R) \times_\theta \mathbb{Z}_n)$ can be reduced to the discrete logarithm problem in $< \Phi >$. In section 4 we show that MOR using $GL(2,R) \times_\theta \mathbb{Z}_n$ is vulnerable to chosen ciphertext attacks if the computational Diffie-Hellman problem in $< \Phi >$ can be solved efficiently. In the final section 5 the impact of the presented attacks on the security of MOR using $GL(2,R) \times_\theta \mathbb{Z}_n$ is discussed and directions for future research are pointed out. The appendix briefly describes how to solve the special conjugacy problem (SCP) in $GL(2,R)$ by solving simultaneous instances of the conjugacy problem (CP) in $GL(2,R)$.

Related Work: The conjugacy problem is considered a hard problem in braid groups. There is no known polynomial time algorithm which solves the decisional or the computational conjugacy problem in braid groups. For a detailed discussion of cryptography on braid groups we refer to [1,3,5]. Other cryptosystems using the conjugation map on matrix groups have been published by Yamamura [9,10]. The systems later were broken by Blackburn and Galbraith [2].

2 Framework and Definitions

Definition 1 (Semi-Direct Product Group). *Let G and H be groups and $\theta : H \to Aut(G)$ be a homomorphism. The set $G \times H = \{(g,h) \mid g \in G, h \in H\}$ together with the multiplication map*

$$(g_1,h_1)(g_2,h_2) = (g_1\theta(h_1)(g_2), h_1h_2)$$

is a group, called the semi-direct product $G \times_\theta H$ of G and H with respect to θ.

Definition 2 (The mapping Inn). *Let G be a group. Then the mapping*

$$\begin{aligned} Inn : G &\to Aut(G) \\ g &\mapsto Inn(g) \end{aligned}$$

is given by $Inn(g)(h) = ghg^{-1}$.

We call $Inn(g)$ an inner automorphism and $Inn(G) = \{Inn(g) \mid g \in G\}$ the inner automorphism group. If G is an abelian group then $Inn(g)$ is the identity map for all $g \in G$ and $Inn(G)$ is trivial. Let $\{\gamma_i\}$ be a set of generators of G. Since $Inn(g)$ is a homomorphism, $Inn(g)$ is totally specified for all $m \in G$ if the values $\{Inn(g)(\gamma_i)\}$ are given.

Definition 3 (center, centralizer). *Let G be a group. The center $Z(G)$ of G is defined as $Z(G) := \{g \in G \mid xg = gx \ \forall x \in G\}$.*
Let $g \in G$. The centralizer $Z(g)$ of g is defined as $Z(g) := \{h \in G \mid hg = gh\}$.
Note that $Z(G) = \bigcap_{g \in G} Z(g)$.

In the appendix the terms ”center” and ”centralizer” are also used for rings resp. ring elements. For a ring R and ring elements $r \in R$ we define $Z(R) := \{r \in R \mid sr = rs \; \forall s \in R\}$ and $Z(r) := \{s \in R \mid rs = sr\}$.
In some cases it may not be clear from the context which structure is referred to, e.g. for $g \in GL(2,R) \subseteq M(2,R)$ the cenralizer $Z(g)$ in the ring $M(2,R)$ may be different from the centralizer $Z(g)$ in the multiplicative group $GL(2,R)$. In this case the corresponding structure is added as an index, e.g. $Z_{M(2,R)}(g) = \{h \in M(2,R) \mid gh = hg\}$ and $Z_{GL(2,R)}(g) = \{h \in GL(2,R) \mid gh = hg\}$.

Definition 4 (Conjugacy Problem). *Let G be a group. For arbitrary $x, y \in G$ the conjugacy problem (CP) is to find $w \in G$ such that $wxw^{-1} = y$.*

Let $w \in G$ be a solution of the instance (x,y) of the CP, i.e. $wxw^{-1} = y$. Then $w \cdot Z(x)$ is the solution set for instance (x,y).

Definition 5 (Special Conjugacy Problem). *For a given $\varphi \in Inn(G)$ the special conjugacy problem is to find an element $g \in G$ satisfying $Inn(g) = \varphi$.*

The solution set for instance $Inn(g)$ of the special conjugacy problem is $g \cdot Z(G)$.
In $GL(2,\mathbb{Z}_p)$ the conjugacy problem is easy. To solve the special conjugacy problem in $GL(2,\mathbb{Z}_p)$ two pairs $(A_1, Inn(A_1))$ and $(A_2, Inn(A_2))$ with $A_1 \notin Z(A_2)$ are needed (see [8] for details). A similar result holds for the group $GL(2,R)$ of invertible matrices over a commutative ring with identity R (see appendix A).

The MOR cryptosystem: MOR is an asymmetric cryptosystem with a random value a as secret and the two mappings $Inn(g)$ and $Inn(g^a)$ (given as $\{Inn(g)(\gamma_i)\}$ and $\{Inn(g^a)(\gamma_i)\}$ for a generator set $\{\gamma_i\}$ of G) as corresponding public key. The encryption process works as follows:

1. Alice expresses the plaintext $m \in G$ as a product of the γ_i.
2. Alice chooses a random $b \in_R \mathbb{Z}_{\mathrm{ord}(Inn(g))}$ and computes $(Inn(g^a))^b$, i.e. $\{(Inn(g^a))^b(\gamma_i)\}$.
3. Alice computes $E = Inn(g^{ab})(m) = (Inn(g^a))^b(m)$.
4. Alice computes $\Phi = Inn(g)^b$, i.e. $\{Inn(g^b)(\gamma_i)\}$.
5. Alice sends the ciphertext $C = (E, \Phi)$ to Bob.

Decryption Process:

1. Bob expresses E as a product of the γ_i.
2. Bob computes Φ^{-a}, i.e. $\{\Phi^{-a}(\gamma_i)\}$.
3. Bob computes $m = \Phi^{-a}(E)$.

The MOR cryptosystem is very similar to the ElGamal cryptosystem [4]. The Diffie-Hellman key establishment protocol is used to fix a common inner automorphism $(Inn(g))^{ab}$. The ciphertext of a message $m \in G$ is the image of m under $Inn(g^{ab}) = (Inn(g))^{ab}$.
In [6] no formal proof of security is given for the MOR system. If the discrete logarithm problem is efficiently solvable in $< Inn(g) >$, then the secret key a can be calculated from $Inn(g), Inn(g^a)$ which are part of the public key. However, knowledge of the secret key is not necessary to attack the MOR cryptosystem for certain non-abelian groups G (see [8] for details).

3 MOR using $GL(2,R) \times_\theta \mathbb{Z}_n$

Let R be a commutative ring with identity and $\Phi : R \to R$ be a (non-trivial) ring isomorphism. Then $GL(2,R) = \{ \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \in M(2,R) \mid a_1a_4 - a_2a_3$ is invertible$\}$ is a (multiplicative) group. A group automorphism ϕ is induced by Φ:

$$\phi : GL(2,R) \to GL(2,R),$$

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \mapsto \begin{pmatrix} \Phi(a_1) & \Phi(a_2) \\ \Phi(a_3) & \Phi(a_4) \end{pmatrix}$$

By setting $\theta(1) = \phi$ we get a homomorphism $\theta : \mathbb{Z}_n \to Aut(GL(2,R))$, i.e.

$$\theta(k) = \phi^k : \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \to \begin{pmatrix} \Phi^k(a_1) & \Phi^k(a_2) \\ \Phi^k(a_3) & \Phi^k(a_4) \end{pmatrix}$$

We now examine MOR using the semi-direct product $GL(2,R) \times_\theta \mathbb{Z}_n$.

The conjugation map in $GL(2,R) \times_\theta \mathbb{Z}_n$:
Let $(x,y),(m_1,m_2) \in G \times_\theta H$. Then:

$$(x,y)(m_1,m_2)(x,y)^{-1} = (x\theta(y)(m_1)\theta(m_2)(x^{-1}), m_2)$$

Applied to the group $G = GL(2,R) \times_\theta \mathbb{Z}_n$ and homomorphism θ we get for $(x,y),(m_1,m_2) \in GL(2,R) \times_\theta \mathbb{Z}_n$:

$$(x,y)(m_1,m_2)(x,y)^{-1} = (x \cdot \phi^y(m_1) \cdot \phi^{m_2}(x^{-1}), m_2)$$

The choice of Φ:
Let $G = GL(2,R) \times_\theta \mathbb{Z}_n$ and Φ, ϕ and θ as defined above. Then

1. $\text{ord}(\Phi) = \text{ord}(\phi)$
2. $\theta(n) = Id_{GL(2,R)} \Leftrightarrow n \equiv 0 \pmod{\text{ord}(\Phi)}$
3. If $(x,y),(x,\hat{y}) \in G$, then $Inn((x,y)) = Inn((x,\hat{y})) \Leftrightarrow y \equiv \hat{y} \pmod{\text{ord}(\Phi)}$
4. The homomorphism θ is well-defined if and only if $\text{ord}(\Phi) \mid n$.

Let $(x,y) \in GL(2,R) \times_\theta \mathbb{Z}_n$ and $(x,y)^{ab} = (\hat{x}, aby \pmod{n})$ for some $\hat{x} \in GL(2,R)$. Then a ciphertext of a message $(m_1,m_2) \in GL(2,R) \times_\theta \mathbb{Z}_n$ looks as follows:

$$Inn((x,y)^{ab})(m_1,m_2) = (\hat{x}\phi^{aby}(m_1)\phi^{m_2}(\hat{x}^{-1}), m_2)$$

The values $a,b,y \in \mathbb{Z}_n$ should have no common divisor with the order of homomorphism ϕ. Otherwise ϕ^{aby} is no generator of the cyclic group $< \phi >$. This reduces the number of possible ciphertexts for a plaintext message $(m_1,m_2) \in GL(2,R) \times_\theta \mathbb{Z}_n$. To avoid this problem, we suggest to choose n prime.

Extracting ϕ^y from $Inn(g)$:
We now show that given an inner automorphism $Inn(g)$ for some $g = (x, y) \in GL(2, R) \times_\theta \mathbb{Z}_n$ the group automorphism ϕ^y can be calculated efficiently.

Step 1: To calculate ϕ^y we make use of the fact that $\Phi^y(0) = 0$ and $\Phi^y(1) = 1$. For a unimodular matrix $m \in GL(2, R)$ (i.e. a matrix with entries only 0 and 1) it follows that $\phi^y(m) = m$ and we get

$$\begin{aligned} Inn(g)(m, 0) = (x, y)(m, 0)(x, y)^{-1} &= (x \cdot \phi^x(m) \cdot \phi^0(x^{-1}), 0) \\ &= (x \cdot m \cdot x^{-1}, 0) \end{aligned}$$

This leads to an instance m, xmx^{-1} of the conjugacy problem in $GL(2, R)$. By solving the two instances $\begin{pmatrix} 0\,1 \\ 1\,0 \end{pmatrix}$, $x \begin{pmatrix} 0\,1 \\ 1\,0 \end{pmatrix} x^{-1}$ and $\begin{pmatrix} 1\,1 \\ 1\,0 \end{pmatrix}$, $x \begin{pmatrix} 1\,1 \\ 1\,0 \end{pmatrix} x^{-1}$ of the conjugacy problem in $GL(2, R)$ simultaneously the special conjugacy problem can be solved and an element $\hat{x} \in GL(2, R)$ with $Inn(x) = Inn(\hat{x})$ can be calculated (see appendix A).

Step 2: For arbitrary $m \in GL(2, R)$ we get

$$Inn(g)(m, 0) = (x, y)(m, 0)(x, y)^{-1} = (x \cdot \phi^y(m) \cdot x^{-1}, 0)$$

Since $Inn(x) = Inn(\hat{x})$ we know that $\hat{x}^{-1} \cdot x \in Z(GL(2, R))$. The image of martix m under ϕ^y can be calculated as follows:

$$Inn(\hat{x}^{-1})(x \cdot \phi^y(m) \cdot x^{-1}) = (\hat{x}^{-1}x) \cdot \phi^y(m) \cdot (\hat{x}^{-1}x)^{-1} = \phi^y(m)$$

Using the same technique the homomorphism ϕ^{ay} can be calculated given $Inn(g^a)$. Since $Inn(g)$ and $Inn(g^a)$ are part of the public key, the two ring homomorphisms ϕ^y and ϕ^{ay} can be calculated efficiently. For the security of MOR using $GL(2, R) \times_\theta \mathbb{Z}_n$ it is necessary that the discrete logarithm problem is hard in $< \phi >$. Otherwise $a \pmod{\mathrm{ord}(\phi)}$ can be calculated which gives partial information of the secret key a.

4 Analysis of MOR using $GL(2, R) \times_\theta \mathbb{Z}_n$

The most time consuming operations in the encryption and decryption process of the MOR cryptosystem are the exponentiations in $< Inn(g) >$. The inner automorphisms are given by the images of the generators $\gamma_1, \ldots \gamma_n$ of the used group G. To calculate $Inn(g^2)(\gamma_i)$, two steps are needed. In the first step $Inn(g)(\gamma_i)$ has to be expressed as a product of the generators γ_i and in the second step the corresponding images $Inn(g)(\gamma_i)$ have to be multiplied. Since 2 (resp. 1) exponentiations in $< Inn(g) >$ have to be calculated during the encryption (resp. decryption) process, the MOR cryptosystem in its basic form is much too inefficient to be of practical interest.
Therefore a variant of MOR has been proposed [6] where the encryption exponent b is used for multiple encryptions. Since the resulting encryption scheme

is deterministic, the authors of [6] recommend to use a probabilistic padding scheme when fixing the encryption exponent.
We now show that MOR using $GL(2,R) \times_\theta \mathbb{Z}_n$ with fixed encryption exponent (even when the probabilistic padding scheme is used) is vulnerable to chosen ciphertext attacks if the computational Diffie-Hellman Problem in $< \phi >$ can be solved (efficiently). From $Inn(g^a)$ (which is part of the public key) and $Inn(g^b)$ (which is part of the ciphertext) the homomorphisms ϕ^{ay} and ϕ^{by} can be computed. Solving the computational Diffie-Hellman problem yields ϕ^{aby}.

Let $c = (c_1, c_2) \in GL(2,R)$ be a given challenge ciphertext of MOR using $GL(2,R) \times_\theta \mathbb{Z}_n$. In a chosen ciphertext attack the attacker is assumed to have access to a decryption oracle. He is allowed to send ciphertexts $\hat{c} \neq c$ to the oracle and gets the corresponding plaintext messages. A cryptosystem is secure against chosen ciphertext attacks if such an attacker is not able to compute the plaintext corresponding to c efficiently.
In our attack we make use of the fact that the encryption function $Inn(g^{ab})$ is an automorphism, i.e. every $d = (d_1, d_2) \in GL(2,R) \times_\theta \mathbb{Z}_n$ is a valid ciphertext of a (maybe unknown) message $m = (m_1, m_2) \in GL(2,R) \times_\theta \mathbb{Z}_n$.
Let $g = (x, y) \in GL(2,R) \times_\theta \mathbb{Z}_n$. Then $(x,y)^{ab} = (\hat{x}, aby \pmod{n})$ for some $\hat{x} \in GL(2,R)$. Ciphertexts of MOR using $GL(2,R) \times_\theta \mathbb{Z}_n$ are of the form

$$d = (d_1, d_2) = (\hat{x} \cdot \phi^{aby}(m_1) \cdot \phi^{m_2}(\hat{x}^{-1}), m_2)$$

The attack consists of two steps. In the first step an $\bar{x} \in GL(2,R)$ with $Inn(\hat{x}) = Inn(\bar{x})$ is computed. This element $\bar{x}$ is used in the second step to decipher the challenge ciphertext c.

Step 1: For every $d_1 \in GL(2,R)$ the value $(d_1, 0) \in GL(2,R) \times_\theta \mathbb{Z}_n$ is a valid ciphertext of the (unknown) message $(m_1, 0) \in GL(2,R) \times_\theta \mathbb{Z}_n$:

$$(d_1, 0) = (\hat{x} \cdot \phi^{aby}(m_1) \cdot \hat{x}^{-1}, 0)$$

Sending $(d_1, 0)$ to the decryption oracle, the attacker gets the corresponding plaintext message $(m_1, 0)$. Since we assumed that the attacker knows ϕ^{aby} he is able to compute $\phi^{aby}(m_1)$. The values $\phi^{aby}(m_1), d_1 = \hat{x} \cdot \phi^{aby}(m_1) \cdot \hat{x}^{-1}$ form an instance of the conjugacy problem in $GL(2,R)$. Repeating this process generates multiple simultaneous instances of the conjugacy problem in $GL(2,R)$ which can be used to solve the special conjugacy problem in $GL(2,R)$ and get a group element $\bar{x} \in GL(2,R)$ with $Inn(\hat{x}) = Inn(\bar{x})$ (see appendix A for details).

The oracle may not answer queries with zero as second component, because $GL(2,R) \times_\theta \{0\}$ is isomorphic to $GL(2,R)$ and the conjugacy problem is efficiently solvable in $GL(2,R)$. In this case the attacker sends queries $(d_1, i), (\hat{d_1}, i) \in GL(2,R) \times_\theta \mathbb{Z}_n$ with the same second component to the decryption oracle:

$$(d_1, i) = (\hat{x} \cdot \phi^{aby}(m_1) \cdot \phi^i(\hat{x}^{-1}), i)$$
$$(\hat{d_1}, i) = (\hat{x} \cdot \phi^{aby}(\hat{m_1}) \cdot \phi^i(\hat{x}^{-1}), i)$$

With the plaintext messages $(m_1, i), (\hat{m_1}, i) \in GL(2, R) \times_\theta \mathbb{Z}_n$ and homomorphism ϕ^{aby} the attacker can compute $\phi^{aby}(m_1) \cdot (\phi^{aby}(\hat{m_1}))^{-1} = \phi^{aby}(m_1 \cdot \hat{m_1}^{-1})$ and $d_1 \cdot (\hat{d_1})^{-1} = \hat{x} \cdot \phi^{aby}(m_1 \cdot \hat{m_1}^{-1}) \cdot \hat{x}^{-1}$ to get an instance of the CP in $GL(2, R)$.

Step 2: Let (p_1, p_2) be the plaintext message encrypted in the challenge ciphertext $c = (c_1, c_2)$. Since $\bar{x} = \hat{x} \cdot z$ for a $z \in Z(GL(2, R))$ we get:

$$\begin{aligned} \bar{x}^{-1} \cdot c_1 \cdot \phi^{c_2}(\bar{x}) &= \bar{x}^{-1} \cdot (\hat{x} \cdot \phi^{aby}(p_1) \cdot \phi^{c_2}(\hat{x})) \cdot \phi^{c_2}(\bar{x}) \\ &= \phi^{aby}(p_1) \cdot z^{-1} \cdot \phi^{c_2}(z) \end{aligned}$$

Only one oracle query is necessary to calculate $z^{-1} \cdot \phi^{c_2}(z)$. The attacker chooses a $c_3 \neq c_1 \in GL(2, R)$ and sends (c_3, c_2) to the oracle. If $\hat{m}$ is the answer of the oracle, the attacker gets $z^{-1} \cdot \phi^{c_2}(z)$ as follows:

$$\begin{aligned} c_3 \cdot (\phi^{c_2}(\bar{x})\phi^{aby}(\hat{m}^{-1})\bar{x}^{-1}) &= (\hat{x}\phi^{aby}(\hat{m})\phi^{c_2}(\hat{x}^{-1})) \cdot (\phi^{c_2}(\bar{x})\phi^{aby}(\hat{m}^{-1})\bar{x}^{-1}) \\ &= \hat{x}\phi^{aby}(\hat{m})\phi^{c_2}(\hat{x}^{-1})\phi^{c_2}(\hat{x}z)\phi^{aby}(\hat{m}^{-1})(\hat{x}z)^{-1} \\ &= z^{-1} \cdot \phi^{c_2}(z) \end{aligned}$$

Now the attacker can compute $\phi^{aby}(p_1)$.

Step 3: If the knowledge of ϕ^{aby} is not sufficient to compute p_1 from $\phi^{aby}(p_1)$, the decryption oracle is used to compute preimages under ϕ^{aby}. To obtain the preimage of $\phi^{aby}(p_1)$ the attacker sends

$$(d_1, 0) = (\bar{x} \cdot \phi^{aby}(p_1) \cdot \bar{x}^{-1}, 0) = (\hat{x} \cdot \phi^{aby}(p_1) \cdot \hat{x}^{-1}, 0)$$

as query to the decryption oracle. The oracle reply equals the wanted preimage. If the oracle does not answer queries with zero as second component the value $\bar{x} \cdot \phi^{aby}(p_1) \cdot \bar{x}^{-1}$ can be expressed as $\bar{x} \cdot \phi^{aby}(p_1) \cdot \bar{x}^{-1} = e_1 \cdot \hat{e_1}^{-1}$ for $e_1, \hat{e_1} \in GL(2, R)$ and (e_1, i) and $(\hat{e_1}, i)$ can be sent to the oracle. If a_1 and $\hat{a_1}$ are the oracle's answers, the desired preimage is $p_1 = a_1 \cdot \hat{a_1}^{-1}$ (see also step 1 for a similar argument).

Using a randomised padding scheme: In [6] the authors propose to use a probabilistic padding scheme when fixing the encryption exponent. The plaintext message $m \in R$ is embedded in $GL(2, R)$ by choosing a random matrix $M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \in GL(2, R)$ with $m_1 = m$. After that the encryption function $Inn(g^{ab})$ is applied to M.
In [8] it has been shown that MOR using $SL(2, \mathbb{Z}_p) \times_\theta \mathbb{Z}_n$ is insecure even if the randomised padding scheme is used: Two pairs consisting of plaintext and corresponding ciphertext are sufficient to calculate $Inn(g^{ab})$. The same techniques can be applied to step 1 of our attack to calculate an element $\bar{x} \in GL(2, R)$ with $Inn(\hat{x}) = Inn(\bar{x})$.

The first part of step 2 also works if the described padding scheme is used, i.e. $\phi^{aby}(p_1) \cdot z^{-1} \cdot \phi^{c_2}(z)$ can be calculated. The second part of step 2 has to be changed slightly: On input (c_3, c_2) the decryption oracle outputs only the $(1,1)$-component of $\hat{m}$. The other entries of matrix $\hat{m}$ are not known to the attacker. Since $Z(GL(2,R)) = \{c \cdot Id \mid c \in R, c \text{ invertible}\}$, the value $z^{-1} \cdot \phi^{c_2}(z)$ is of the form $z^{-1} \cdot \phi^{c_2}(z) = \begin{pmatrix} r & 0 \\ 0 & r \end{pmatrix}$ for an invertible $r \in R$. In particular $z^{-1} \cdot \phi^{c_2}(z) \in Z(GL(2,R))$. For $\hat{m} = \begin{pmatrix} \hat{m}_1 & \hat{m}_2 \\ \hat{m}_3 & \hat{m}_4 \end{pmatrix}$ we get

$$\begin{aligned}
\bar{x}^{-1} \cdot c_3 \cdot \phi^{aby}(\bar{x}) &= (z^{-1}\hat{x}^{-1}) \cdot (\hat{x}\phi^{aby}(\hat{m})\phi^{c_2}(\hat{x}^{-1})) \cdot (\phi^{c_2}(\hat{x}z)) \\
&= \phi^{aby}(\hat{m}) \cdot z^{-1} \cdot \phi^{c_2}(z) \\
&= \begin{pmatrix} r \cdot \hat{m}_1 & r \cdot \hat{m}_2 \\ r \cdot \hat{m}_3 & r \cdot \hat{m}_4 \end{pmatrix}
\end{aligned}$$

The value $\hat{m}_1$ can be obtained by sending (c_3, c_2) to the decryption oracle. If r cannot be calculated given $\hat{m}_1$ and $r \cdot \hat{m}_1$ this process has to be repeated with a different value c_3.

Step 3 also works when the randomised padding scheme is used but has to be carried out for every single component, i.e. to compute the preimage of $\phi^{aby}(p_1) = \begin{pmatrix} \Phi^{aby}(p_{11}) & \Phi^{aby}(p_{12}) \\ \Phi^{aby}(p_{13}) & \Phi^{aby}(p_{14}) \end{pmatrix}$ step 3 is used to find preimages of $d_i \in GL(2,R)$, $1 \leq i \leq 4$, where the $(1,1)$-component of d_i equals $\Phi^{aby}(p_{1i})$.

5 Conclusion

We showed that MOR using $GL(2,R) \times_\theta \mathbb{Z}_n$ with fixed encryption exponent is vulnerable to chosen ciphertext attacks if the computational Diffie-Hellman Problem is easy in $< \Phi >$. The presented attacks still work if the randomised padding scheme of [6] is used. They do not work if the encryption exponent b is randomly chosen for every plaintext to be encrypted. However, in this case two exponentiations in $< Inn(g) >$ have to be calculated during the encryption and one during the decryption process. The resulting cryptosystem is too inefficient to be of practical interest.

Our results show that the hardness of the discrete logarithm problem (DLP) in $< \Phi >$ is essential for the security of all modes of MOR (even when the encryption exponent b is chosen randomly and independently for every plaintext to be encrypted). The DLP in $< \Phi >$ is much easier than the DLP in $< Inn(g) >$ (which has to be solved to calculate the secret key given the public key). It may be more appropriate to use a variant of the ElGamal cryptosystem [4] using the cyclic group $< \Phi >$. The resulting cryptosystem would be provable secure and more efficient than MOR using $GL(2,R) \times_\theta \mathbb{Z}_n$.

All attacks are generic attacks, i.e. they work for every ring R and every homomorphism Φ. For certain choices of R and Φ there may be even stronger attacks. It is a task for future reserach to find a non-abelian group suitable for the use with the MOR cryptosystem.

References

1. I. Anshel, M. Anshel, D. Goldfeld, "An Algebraic Method for Public-Key Cryptography", Mathematical Research Letters, 6 (1999), pp. 287-291
2. S. Blackburn, S. Galbraith, "Cryptanalysis of two cryptosystems based on group action", Advances in Cryptology - Asiacrypt 1999, LNCS 1716, pp. 52-61
3. P. Dehornoy, "Braid-based cryptography", Preprint, University of Caen, 2003, http://matin.math.unicaen.fr/~dehornoy/papers.html
4. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, Volume 31, 1985, pp. 469-472
5. K. H. Koo, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang, C. Park, "New Public-Key Cryptosystem Using Braid Groups", Advances in Cryptology - Crypto 2000, LNCS 1880, pp. 166-183
6. S.-H. Paeng, K.-C. Ha, J. H. Kim, S. Chee, C. Park, "New Public Key Cryptosystem Using Finite Non Abelian Groups", Advances in Cryptology - Crypto 2001, LNCS 2139, pp. 470-485
7. S.-H. Paeng, D. Kwon, K.-C. Ha, J. H. Kim, "Improved public key cryptosystem using finite non abelian groups", IACR EPrint-Server, Report 2001/066, http://eprint.iacr.org/2001/066
8. C. Tobias, "Security Analysis of the MOR Cryptosystem", 6th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2003, LNCS 2567, pp. 175-186
9. A. Yamamura, "Public key cryptosystems using the modular group", 1st International Public Key Cryptography Conference PKC 1998, LNCS 1431, pp. 203-216
10. A. Yamamura, "A functional cryptosystem using a group action", 4th Australian Information Security and Privacy Conference, ACISP 1999, LNCS 1587, pp. 314-325

A The Special Conjugacy Problem in $GL(2, R)$

Let $Inn(g) : GL(2, R) \rightarrow GL(2, R)$ be a public inner automorphism. We assume that $Inn(g)$ is given as a black box, i.e. an attacker is able to calculate images under $Inn(g)$ but does not know the used $g \in GL(2, R)$. This approach assures that our calculations are independent of the presentation of $Inn(g)$. We now show that the special conjugacy problem is efficiently solvable in $GL(2, R)$.

Let $B, C, X \in GL(2, R)$ and $B, XBX^{-1} = \hat{B} = \begin{pmatrix} \hat{b_1} & \hat{b_2} \\ \hat{b_3} & \hat{b_4} \end{pmatrix}$ and $C, XCX^{-1} = \hat{C} = \begin{pmatrix} \hat{c_1} & \hat{c_2} \\ \hat{c_3} & \hat{c_4} \end{pmatrix}$ be two simultaneous instances of the conjugacy problem in $GL(2, R)$.

Let $\hat{X} \in GL(2, R)$ be a solution of these two instances. Then $\hat{X} = Z \cdot X$ with $\begin{pmatrix} z_1 & z_2 \\ z_3 & z_4 \end{pmatrix} = Z \in Z(\hat{B}) \cap Z(\hat{C})$. By comparing the components of $Z \cdot \hat{B}$, $\hat{B} \cdot Z$ and $Z \cdot \hat{C}$, $\hat{C} \cdot Z$ we get:[1]

[1] Since $\hat{X}$ could also be expressed as $\hat{X} = X \cdot \hat{Z}$ for a $\hat{Z} \in Z(B) \cap Z(C)$, the following paragraph is also true if $\hat{b_i}$ and $\hat{c_i}$ are replaced by b_i and c_i. In particular $B \in Z(C) \Leftrightarrow \hat{B} \in Z(\hat{C})$.

- $z_2(\hat{c}_3\hat{b}_2 - \hat{b}_3\hat{c}_2) = 0$ and $z_3(\hat{c}_3\hat{b}_2 - \hat{b}_3\hat{c}_2) = 0$
- $z_2(\hat{c}_2(\hat{b}_1 - \hat{b}_4) - \hat{b}_2(\hat{c}_1 - \hat{c}_4)) = 0$ and $z_3(\hat{c}_2(\hat{b}_1 - \hat{b}_4) - \hat{b}_2(\hat{c}_1 - \hat{c}_4)) = 0$
- $z_2(\hat{c}_3(\hat{b}_1 - \hat{b}_4) - \hat{b}_3(\hat{c}_1 - \hat{c}_4)) = 0$ and $z_3(\hat{c}_3(\hat{b}_1 - \hat{b}_4) - \hat{b}_3(\hat{c}_1 - \hat{c}_4)) = 0$

If $\hat{c}_3\hat{b}_2 = \hat{b}_3\hat{c}_2$, $\hat{c}_2(\hat{b}_1 - \hat{b}_4) = \hat{b}_2(\hat{c}_1 - \hat{c}_4)$ and $\hat{c}_3(\hat{b}_1 - \hat{b}_4) = \hat{b}_3(\hat{c}_1 - \hat{c}_4)$, then $\hat{B} \in Z(\hat{C})$. Therefore, were $\hat{B}, \hat{C} \in GL(2, R)$ chosen such that $\hat{B} \notin Z(\hat{C})$, one of the equations has to be false and z_2 and z_3 are zero divisors.

If $\hat{B}, \hat{C} \in GL(2, R)$ where chosen such that $\hat{c}_3\hat{b}_2 - \hat{b}_3\hat{c}_2$, $\hat{c}_2(\hat{b}_1 - \hat{b}_4) - \hat{b}_2(\hat{c}_1 - \hat{c}_4)$ or $\hat{c}_3(\hat{b}_1 - \hat{b}_4) - \hat{b}_3(\hat{c}_1 - \hat{c}_4)$ is no zero divisors it further follows that $z_2 = z_3 = 0$. If one of the ring elements $\hat{b}_2$, $\hat{b}_3$, $\hat{c}_2$ or $\hat{c}_3$ is no zero divisor, then $Z = \begin{pmatrix} z_1 & 0 \\ 0 & z_1 \end{pmatrix}$ for a $z_1 \in R$. Since $Z \cdot M = M \cdot Z$ for all $M \in M(2, R)$, we get that $Inn(X) = Inn(\hat{X})$, i.e. $\hat{X} \in GL(2, R)$ is a solution of the instance $Inn(X)$ of the special conjugacy problem in $GL(2, R)$.

We now show that a simultaneous solution of these two instances can be calculated efficiently. The equations $XBX^{-1} = \hat{B}$ and $XCX^{-1} = \hat{C}$ are equivalent to $XB = \hat{B}X$ and $XC = \hat{C}X$. If $B \notin Z(C)$ this yields to a system of three linear equations. In the presented attack in section 4 the elements $\hat{B}, \hat{C} \in GL(2, R)$ can be chosen freely. If $\hat{b}_3$ is invertible, the obtained system of linear equations is equivalent to:

$$\begin{array}{rcrcl} x_1 + & \frac{\hat{b}_4 - b_1}{\hat{b}_3} \cdot x_3 & - & \frac{b_3}{\hat{b}_3} \cdot x_4 & = 0 \\ x_2 - & \frac{b_2}{\hat{b}_3} \cdot x_3 & + & \frac{\hat{b}_4 - b_4}{\hat{b}_3} \cdot x_4 & = 0 \\ & \multicolumn{3}{r}{(\hat{c}_4 - c_1 - \hat{c}_3 \cdot \frac{\hat{b}_4 - b_1}{\hat{b}_3}) \cdot x_3 - (c_3 - \hat{c}_3 \cdot \frac{b_3}{\hat{b}_3}) \cdot x_4} & = 0 \end{array}$$

For arbitrary $r \in R$ this system is solved by $x_1 = k_1 \cdot r$, $x_2 = k_2 \cdot r$, $x_3 = k_3 \cdot r$ and $x_4 = k_4 \cdot r$ where $k_4 = \hat{c}_4 - c_1 - \hat{c}_3 \cdot \frac{\hat{b}_4 - b_1}{\hat{b}_3}$, $k_3 = (c_3 - \hat{c}_3 \frac{b_3}{\hat{b}_3}) \cdot k$, $k_2 = \frac{b_2}{\hat{b}_3} \cdot k_3 - \frac{\hat{b}_4 - b_4}{\hat{b}_3} \cdot k_4$ and $k_1 = \frac{b_3}{\hat{b}_3} \cdot k_4 - \frac{\hat{b}_4 - b_1}{\hat{b}_3} \cdot k_3$.

If either $\hat{b}_3 c_3 - \hat{c}_3 b_3$ or $\hat{b}_3(\hat{c}_4 - c_1) - \hat{c}_3(\hat{b}_4 - b_1)$ is no zero divisor, $\begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix} \in GL(2, R)$ and $\begin{pmatrix} rk_1 & rk_2 \\ rk_3 & rk_4 \end{pmatrix} \neq \begin{pmatrix} \hat{r}k_1 & \hat{r}k_2 \\ \hat{r}k_3 & \hat{r}k_4 \end{pmatrix}$ for $r, \hat{r} \in R$ with $r \neq \hat{r}$, i.e. we get $\mid R \mid$ distinct solutions. In this case we further know that $\begin{pmatrix} rk_1 & rk_2 \\ rk_3 & rk_4 \end{pmatrix} \in GL(2, R)$ if and only if $r \in R$ is no zero divisor.

Since $X \in GL(2, R)$, the equation $XB = \hat{B}X$ is equivalent to $\hat{B} = XBX^{-1}$. For an element $\hat{X} \in M(2, R)$ with $\hat{X}B = \hat{B}\hat{X}$ we get that $(X^{-1}\hat{X})B = B(X^{-1}\hat{X})$ holds, i.e. $\hat{X} = X \cdot Z$ with $Z \in Z_{M(2,R)}(B)$.

Thus, the simultaneous solutions (in $M(2, R)$) of the equations $XB = \hat{B}X$ and $XC = \hat{C}X$ are of the form $Z \cdot X$ where $Z \in Z_{M(2,R)}(\hat{B}) \cap Z_{M(2,R)}(\hat{C})$. If $\hat{B}, \hat{C} \in GL(2, R)$ were chosen such that $Z_{M(2,R)}(\hat{B}) \cap Z_{M(2,R)}(\hat{C}) = Z(M(2, R))$, there are $\mid Z(M(2, R)) \mid = \mid R \mid$ many solutions, i.e. all solutions are given by $x_1 = k_1 \cdot r$, $x_2 = k_2 \cdot r$, $x_3 = k_3 \cdot r$ and $x_4 = k_4 \cdot r$ with $r \in R$.

Open Secure Infrastructure to control User Access to multimedia content

Carlos Serrão[1], Gregor Siegert[2]

[1] Adetti/ISCTE, Ed. ISCTE – Av. Das Forças Armadas, 1600-082 Lisboa, Portugal
Carlos.Serrao@iscte.pt
[2] Avanti Communications, 28-30 Hoxton Square, London N16NN United Kingdom
Gregor.Siegert@avanti-communications.com

Abstract. This paper describes the OpenSDRM security, based on an open-source framework developed for the IST project MOSES, OpenSDRM is used to control the multimedia content consumption in conjunction with the new MPEG-4 IPMPX proposed standard. This architecture, composed by several building blocks, protects the content flow from creation to final user consumption on a specific device.

1 Introduction

OpenSDRM deploys a secure and distributed DRM solution for content rights protection that can be applied for publishing and trading of digital multimedia content. This architecture started from the OPIMA international specifications [1], MPEG-4 IPMP Extensions [2] and the emerging MPEG-21 IPMP architecture [6]. OpenSDRM is being developed primarily in the scope of the MOSES project, an EC project joining companies from all over Europe, implementing the new MPEG-IPMP Extensions framework and at the same time developing business models and applications for secure content exchange between embedded devices [3, 5]. This solution is composed of several optional elements covering the content distribution value chain, from content production to content usage. It covers several major aspects of the content distribution and trading: content production, preparation and registration, content, interactive content distribution, content negotiation and acquisition, strong actors and user's authentication and conditional visualization/playback [3]. Figure 1 shows the architecture that will be explained in the next section in greater detail. The communication between the components will take place within insecure networks. This introduces special needs regarding the security of this communication. The concept behind the platform is the existence of two security layers. A first security layer established at the communication level, which provides the necessary secure and authenticated communication medium to components to communicate with each other and a second layer established at the application level, ensuring the security, integrity, authentication and non-repudiation mechanisms needed by the different components.

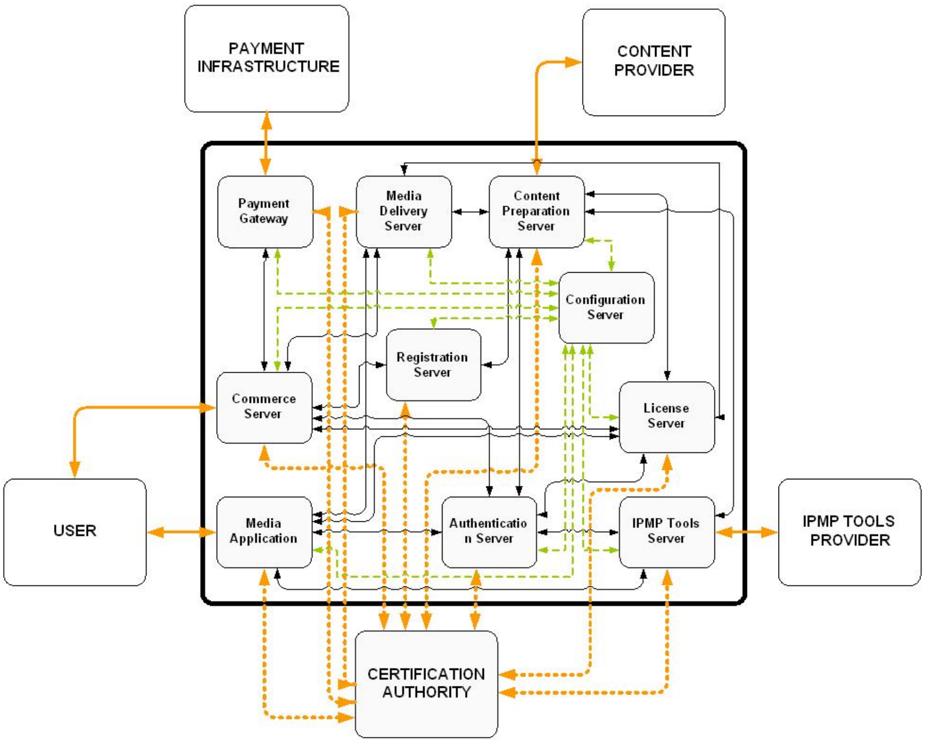


Fig. 1 - OpenSDRM platform architecture composed of several external (User, Payment Infrastructure, Content Provider, IPMP Tools Provider, Certification Authority) and internal components (Payment Gateway (PGW), Media Delivery Server (MDS), Content Preparation Server (CPS), Commerce Server (COS), Registration Server (RGS), Configuration Server (CFS), License Server (LIS), Media Application (MPL), Authentication Server (AUS) and IPMP Tools Server (ITS)).

1.1 Server Components Certification and Registration on OpenSDRM

To establish the secure transport layer, the software components use SSL/TLS protocol. Each of the servers, have a X.509 certificate issued by a Certification Authority (CAU). The CAU can be operated internally by OpenSDRM itself or can be an external and commercial one. OpenSDRM establishes an underlying secure and authenticated transport channel that allows messages to flow from component to component securely. The process works this way: (a) Each component computes a key pair (public and private) , K_{pub}^{Server}, K_{priv}^{Server}, using the RSA algorithm and create Certificate Signing Request (CSR) using its public key and some additional information sending it after to the CAU; (b) The CAU verifies the CSR validity and issues the X.509 SSL certificate to the component, $Cert_{X.509}^{Server}$; (c) The certificate is installed and the components can use SSL/TLS to communicate, establishing the secure transport layer. The architecture requires both components and Users to be registered, in order to establish the Application/Transaction level security. Concerning components those are registered on OpenSDRM AUS. In order to complete this process the following steps are necessary, during the installation of each of the components: (a) Each component computes a key-pair (1024 bit length RSA keys, but higher key lengths are also possible): $Kpub^{Component}$, $Kpriv^{Component}$ (respectively the public and private keys); (b) The component administrator selects a login and a password, and ciphers the $Kpriv^{Component}$, using AES, with the key (K_{AES}) deduced from the hash of the concatenation of the login and password selected: K_{AES} := MD5(login+password). The ciphered component private key gets then protected from unauthorized usage: $K_{AES}[Kpriv^{Component}]$; (c) The component then connects to the AUS and sends some registration information together with the $Kpub^{Component}$. AUS verifies the information sent by the component, validates and registers it, and issues a certificate for the component: $Cert_{AUS}^{Component}$. This certificate is returned to the component. With these component certificates, each of the components will be able to establish trust relationships among them and sign and authenticate all the transactions – this establishes then the Application Level security.

1.2 User's registration on the OpenSDRM platform

In OpenSDRM three components interact directly with external users/entities – MPL, CPS and ITS. These users, respectively Content Users, Content Providers and IPMP Tools Providers are registered on the platform, through the AUS. Content Providers and IPMP Tools Providers, subscribe respectively on the CPS and ITS, relying on the registration and authentication functionalities of the AUS. Therefore, when a new user subscribes, it provides some personal information, a login and password and requests the registration. The following processes can be described like this: (a) The components (ITS and CPS) gather the new registrant information (Info) and request the registration of a new user on the AUS; (b) The components build a new message: $SignKpriv^{Component}\{Component_{ID}, Info\}$. This message is send to AUS; (c) AUS verifies and validates the message, registering the new User and returning a unique $User_{ID}$ to the component. Registering a Content User is a more complex process. This is due to the fact that while both Content Providers and IPMP Tool Providers have their information stored on remote servers, Content Users rely on their own platforms to store their data. In order to provide some additional degree of security, OpenSDRM provides a digital wallet, capable of storing sensitive information such as cryptographic data and licenses in a secure way. The process to register new Content Users can be described in the following steps: (a) When the user runs the wallet for the first time, it creates the User a RSA key pair ($Kpriv^{User}$, $Kpub^{User}$) and asks the user to enter a login and a password; (b) Using the entered login and password, it creates the secure repository master key: K_{AES} = MD5(login+password), and stores sensitive information (Info) on it: $K_{AES}[Info]$; (c) The wallet asks the user to enter some personal data ($Person_{Data}$) and also some payment data (Pay_{Data}) used to charge the user for any commercial content usage; (d) The wallet requests the AUS to register a new User, sending all the information ciphered with the AUS $Kpub^{AUS}$: $Kpub^{AUS}[Person_{Data}, Pay_{Data}, KPriv^{User}, Kpub^{User}]$; (e) AUS receives the data, deciphers it and registers the User. AUS responds to the Wallet with a new certificate generated for the User: $\text{Cert}_{\text{AUS}}^{\text{User}}$, containing among other information the unique identifier of the User, its public key, the identification of the AUS its signature; (f) The wallet stores all the relevant information on the secure repository: $\text{K}_{\text{AES}}[\text{Cert}_{\text{AUS}}^{\text{User}}]$.

1.3 Components message exchange

The process for the components to exchange messages and to verify the authenticity and validity of such messages is composed of the following steps: (a) The sender component (CSender) composes a message using the following syntax: $\text{SignKpriv}^{\text{CSender}}\{\text{CSender}_{\text{ID}}$, Payload, $\text{Cert}_{\text{AUS}}^{\text{CSender}}\}$; (b) The receiver component (CReceiver) receives the message and verifies the trust on the message. This trustability is assured in the following way: (a) CReceiver gets $\text{Cert}_{\text{AUS}}^{\text{CSender}}$ and checks if it was issued by a AUS in which CReceiver trusts; (b)This verification can be conducted if CReceiver has also a certificate issued by AUS: $\text{Cert}_{\text{AUS}}^{\text{CReceiver}}$; (c) After the trust is established, the message signature can be verified and validated and CReceiver can trust its contents, and also in the component who has sent this message; (d) CReceiver can then process the message payload and return its results for the CSender; (e) CReceiver returns the following message to CSender: $\text{SignKpriv}^{\text{CReceiver}}\{\text{CReceiver}_{\text{ID}}$, Results, $\text{Cert}_{\text{AUS}}^{\text{CReceiver}}\}$.

1.4 Payment information and services

Payment of content usage is one of the questions that OpenSDRM also deals and incorporates mechanisms for payment. To provide this functionality a direct trust relationship must be established between the COS and the PGW. Therefore the COS needs to subscribe a PGW. The process to subscribe a PGW can be described as the following: (a) The COS connects to the AUS and asks the AUS which are the PGW available on the system. COS sends $\text{SignKpriv}^{\text{COS}}\{\text{COS}_{\text{ID}}$, RequestAvailablePGWs$\}$ to AUS; (b) AUS verifies the message, and returns an answer to the COS: $\text{SignKpriv}^{\text{AUS}}\{$<ListOfAvailablePGWs, $\text{Cert}_{\text{AUS}}^{\text{PGW}}$>$\}$; (c) The COS selects one available PGW and sends to it a subscription request: $\text{SignKpriv}^{\text{COS}}\{\text{AUS}_{\text{ID}}$, SubscribePGW, $\text{Cert}_{\text{AUS}}^{\text{COS}}\}$; PGW receives the request from the COS, validates its request and subscribes the COS. Therefore, this PGW will be used to validate and process payments used by a given User. Using the payment service provided in OpenSDRM involves two steps: validating the payment instrument and capturing the payment. Validating the payment instrument allows the COS to trust the payment method supplied by the user, and that the transaction can be conducted without problems. Validating the payment involves the following steps: (a) The COS sends information about the payment details, namely information about the User order and the price to pay for it, to AUS: $\text{SignKpriv}^{\text{COS}}\{\text{COS}_{\text{ID}}$, U_{ID}, PGW_{ID}, PayData$\}$; (b) AUS verifies and validates the COS request and checks the UID in order to retrieve the appropriate payment method choose by the User upon registration on the AUS. This data is ciphered with the public key of the PGW: $\text{Kpub}_{\text{PGW}}[\text{PaymentClearence}_{\text{U}}]$; (c) The AUS returns this information for the COS, signing it: $\mathrm{SignKpriv_{AUS}}$\{$\mathrm{Kpub_{PGW}}$[$\mathrm{PaymentClearence_U}$]\}; (d) This information is then passed by the COS to the PGW, requesting it to validate the payment transaction: $\mathrm{SignKpriv_{COS}}$\{$\mathrm{COS_{ID}}$, $\mathrm{Kpub_{PGW}}$[$\mathrm{PaymentClearence_U}$]\}; (e) PGW validates the message and deciphers the User payment clearance, using this information to communicate to the corresponding Payment Infratructure, validating it. After, the PGW returns the result of the payment validation to the COS: $\mathrm{SignKpriv_{COS}}$\{$\mathrm{PGW_{ID}}$, $\mathrm{Transaction_{ID}}$\}; This concludes the payment method validation on the PGW, assuring the COS that the services supplied to the User will be charged. The second step in the payment procedure involves the payment capture. This process requires that first a payment capture has occurred and second that the COS owns a $\mathrm{Transaction_{ID}}$. The capture process can be described in the following: (a) COS sends a message to PGW: $\mathrm{SignKPriv_{COS}}$\{$\mathrm{COS_{ID}}$, $\mathrm{Transaction_{ID}}$\}; (b) PGW validates the message and verifies the $\mathrm{Transaction_{ID}}$, in order to evaluate if that transaction is in fact pending, and processes the payment; (c) PGW returns and a result status to the COS: $\mathrm{SignKPriv_{PGW}}$\{$\mathrm{PGW_{ID}}$, $\mathrm{Transaction_{ID}}$, Result\}.

1.4 License Production, download and expiry

One of the major functionalities of the OpenSDRM platform resides on the fact that it can control the way the Users access and use the content protected by the platform. This process is ensured by the production of licenses. These are later applied on the content of the user on the client player by the appropriate set of IPMP tools. These licenses are produced and stored securely by the LIS, according to the choices made by the User and after the payment has been performed. The process can be described in the following steps: (a) The User selects a set of available conditions, that allow him to define the usage conditions (rights) of the content the User wants to access; (b) COS sends a message to the LIS, requesting the production of a new license, for a specific content, and for a given User: $\mathrm{SignKpriv_{COS}}$\{$\mathrm{U_{ID}}$, $\mathrm{Content_{ID}}$, LicenseConditions, $\mathrm{Cert_{AUS}}^{\mathrm{COS}}$\}; (c) LIS receives the request, verifies it and validates it. LIS generated the license using the appropriate language and parameters, contacting after the AUS for ciphering the license data for the User: $\mathrm{SignKpriv_{LIS}}$\{License\}; (d) AUS receives the data, retrieves the $\mathrm{Kpub_{User}}$ and ciphers the received data: $\mathrm{Kpub_U}$[License], returning it afterwards to the LIS: $\mathrm{SignKpriv_{AUS}}$\{$\mathrm{Kpub_{User}}$[License]\}; (e) LIS stores $\mathrm{Kpub_{User}}$[License]. When the User tries to access the content on the client side the player verifies that a license is needed to access the content. The player contacts the wallet to try to obtain the required licenses and corresponding keys to access the content. This process can be described in the following steps: (a) The player contacts the wallet to obtain the license for the ContentID and UserID; (b) The wallet checks on its secure repository if a license for that specific ContentID is already there. If that is true than this license is returned for the player in order for the content to be deciphered and accessed, controlled by a set of IPMP tools. If the wallet doesn't contain the license, it will request it from the LIS: $\mathrm{SignKpriv_U}$\{$\mathrm{Cert_{AUS}}^{\mathrm{User}}$, $\mathrm{Content_{ID}}$\}; (c) LIS receives the data, validates it and retrieves the license from the database, passing it to the wallet: $\mathrm{SignKpriv_{LIS}}$\{$\mathrm{Kpub_{User}}$[License], $\mathrm{Cert_{AUS}}^{\mathrm{LIS}}$\}; (d) The wallet receives the data from the LIS, validates the message and deciphers the license that is passed to the player. Also the license is stored on the wallet secure repository for future accesses. The downloaded license is kept in the LIS for later crash recovery in an event of failure and later expiration checks. Depending on the rights specified, a license will eventually expire. Rights such as a play count or a validity period may restrict the access to content to a certain number of times or to a certain time frame. The state of the license is maintained within the digital wallet. Upon expiration, for example when a play count reaches zero, the wallet automatically checks at the LIS for a new license for that particular content. If there is no license available and the user wants to continue with the consumption of the content he has purchase a new License as described before. The LIS also applies an internal checking algorithm to manage the state of its licenses. Licenses that expired will be removed from the LIS.

2 Conclusions

This paper presented and discussed the security aspects of an open platform for the multimedia content IPR. The focus was mostly in the description of the security protocol and secure message exchanging which is established among the different components [4]. OpenSDRM relies on text-based communication. Therefore it defines a two layered security protocol [3]. OpenSDRM, contrarily to the normal operation followed by other DRM solutions, addresses DRM using an open approach, following open standards and open-source software. Finally, it is important to stress the fact that although OpenSDRM is mostly an open-standards and open-source based solution, this doesn't prevent that some parts of the system may be closed. An example of this is the fact that the authoring tools used to protect the content itself may be closed. Protection tools, such as watermarking algorithms and specific scrambling or encryption algorithms may be closed, although they are used on an open environment such as OpenSDRM.

References

1. Chiariglione, L., “Intellectual Property in the Multimedia Framework”, Management of Digital Rights, Berlin, 2000
2. Jack Lacy, Niels Rump, Panos Kudumakis, "MPEG-4 Intellectual Property Management & Protection (IPMP) - Overview & Applications Document", ISO/IEC JTC1/SC29/WG11/N2614, 1998
3. Gregor Siegert, Carlos Serrão, "An Open-Source Approach to Content Protection and Digital Rights Management in Media Distribution Systems", ICT Conference 2003, Copenhagen December 2003
4. “Digital Rights: Background, Systems, Assessment”, Commission Staff Working Draft, Commision of the European Communities, 2002
5. Open Mobile Alliance “Generic Content Download Over The Air Specification”, v1.0 December 2002
6. Multimedia Description Schemes (MDS) Group, "MPEG-21 Rights Expression Language WD V3", ISO/IEC JTC1/SC29/WG11/N4816, 2002

Fair Trading Protocol With Off-line Anonymous Credit Card Payment

Weiliang Zhao[1], Vijay Varadharajan[1,2], and George Bryan[1]

[1] Center for Advanced Systems Engineering
University of Western Sydney,
Locked Bag 1797
Penrith South DC, NSW 1797, Australia
{wzhao, g.bryan}@cit.uws.edu.au
[2] Department of Computing,
Macquarie University,
NSW 2109, Australia
vijay@ics.mq.edu.au

Abstract. A fair trading protocol with off-line anonymous credit card payment is proposed in this paper. The fair trading protocol provides an overall solution for a trading process with off-line anonymous credit card payment. The fairness is achieved for both the involved client and merchant. The client information about credit card is anonymous in the protocol. The proposed protocol is based on the general optimistic protocols for fair exchange with an off-line Trusted Third Party (TTP). The financial institution for credit card service can be off-line in the fair trading protocol. The TTP and the financial institution for the credit card service are not involved in normal transactions and the running cost will be reduced.

1 Introduction

With the exploding growth of electronic commerce on the Internet, the issue of fairness [1, 2] is becoming increasingly more important. Fair exchange protocols have already been broadly used for applications such as electronic transactions [3, 4], electronic mails [5, 6], and contract signing [7]. The fairness is one of critical issues in on-line transactions and related electronic payment systems. Many electronic payment systems have been proposed for providing different levels of security to financial transactions, such as iKP [8], SET [9], NetBill[10] and NetCheque [11]. In a normal electronic commerce transaction, there is always a payer and a payee to exchange money for goods or services with each other. At least one financial institution, normally a bank, should be present in the payment system. The financial institution will play the role of issuer for the payer and the role of acquirer for the payee. An electronic payment system must enable an honest payer to convince the payee of a legitimate payment and prevent a dishonest payer from making other unsuitable behaviors. At the same time, some additional security requirements may be addressed based on the nature of trading processes and trust assumptions of the system. Payer, payee and the financial institution have different interests and the trust between two parties should be as little as possible. In electronic commerce, the payment happens over an open network, such as the Internet, the issue of fairness must be carefully addressed. There is no fairness for involved parties in the existing popular payment protocols. One target of this paper is to address the fairness issue in the credit card payment process. In the existing credit card protocols, the financial institution that provides the credit card service plays a role of on-line authority and will be actively involved in a payment. To avoid the involvement of financial institution in normal transactions and reduce running costs, some credit card based schemes with off-line financial authority has been proposed [12]. Another target of this paper is to avoid the on-line financial institution for credit card service in the normal transactions.

In this paper, we propose a fair trading protocol with off-line anonymous credit card payment. The protocol addresses the fairness and privacy of the trading process and its associated payment. The credit card is anonymous and an on-line credit card service from a financial institution is not necessary during the processing of a payment. The TTP and financial institution for credit card can be both off-line, the proposed protocol has better availability and reliability and is more efficient than other solutions with more on-line components. The technique of proof of equivalence of discrete logarithm to discrete log-logarithm [13] is the essential tool in the constructing of our fair trading protocol. In section 2, the electronic payment with off-line anonymous credit card is discussed. In section 3, we propose a fair exchange protocol with off-line anonymous credit card based payment. Finally, section 4 concludes the paper with some final remarks.

2 Electronic Payment With Anonymous Off-line Credit Card

Credit card payment is currently the most popular of all on-line payment methods. There are at least three parties involved in this kind of payments: Client, Merchant and Bank. The client is the buyer or service user who will make the payment. The merchant is the goods or service provider who will receive the payment. The bank is the financial institution that provides credit card service and guarantees the transfer of money value from the client to the merchant. The bank acts as the issuer of credit cards to clients and acquirer of payment records from merchants. For one payment, the issuer and acquirer can be same or different, clearing between the issuer and the acquirer will be done using existing financial networks. There is an on-line financial authority in the existing electronic credit card protocols [8–11]. The authors in [12] have proposed a credit based payment scheme in which the financial institution is not necessary on-line. Merchant can ensure the authenticity of the credit cards without the help of an on-line authority organization. Firstly, the client applies for a digital credit card from the bank. After the credit check, if the client is approved to have it, the digital credit card is delivered to the client through a secure channel. The credit information of the client is anonymous with the technique of no-interactive equality proof [16].

The digital credit card contains at least the following information:

- client's ID
- $h_i = g_i^x \bmod q$, $i = 1, 2, \ldots, l$, where $g_i \in Z_p^*$ are the common generators, x contains the credit card number, PIN number, other confidential information and salt.
- credit amount A

- expiry date E

The digital credit card token is of the form $\mathcal{C} = \; < C, h_1, h_2, \cdots, h_l, E, A >_{skb}$. It has the signature of the bank. If a client sends his digital credit card to a merchant, the merchant can know the credit amount, the expiry date and can check the signature of the bank but can not know the credit card number and PIN number. The client must prove to the merchant that he knows the secret (credit card number, PIN number and other confidential information in the credit card) without revealing the secret to the merchant. Using the technology of equality proof of knowledge, the client chooses a random number r, $r \in Z_p^*$ to compute $a_i = g_i^r$ mod p for all $i = 1, 2, \ldots, l$. The pair $\{c, z\}$ is calculated as:

$$c = H(g_1||g_2||\cdots||g_l||a_1||a_2||\cdots||a_l||h_1||h_2||\cdots||h_l),$$
$$z = cx + r \bmod p.$$

The client will send $\{c, z, p, g_1, \ldots, g_l, a_1, \ldots, a_l, h_1, \ldots, h_l\}$ to the merchant and the merchant can use the following equation to check the validity of the digital credit card.

$$g_i^z \overset{?}{=} h_i^c a_i \bmod p.$$

In any case, the merchant has the option to get confirmation from the authority organization for higher level of assurance. The credit card is anonymous and the financial authority is normally off-line.

3 Fair Trading Protocol with Off-line Anonymous Credit Card Payment

Based on the well-known optimistic protocol for fair exchange[14, 15, 17], we will propose a generic fair trading protocol with off-line anonymous credit card payment. The proposed protocol is an overall solution with the off-line TTP and off-line financial institution for credit card service. The credit information of the client is anonymous in the protocol.

3.1 Notations

Here we give the general notations which will be used in the description of the fair trading protocol.

(1) Parties:

- C: Client
- M: Merchant
- TTP: Trusted Third Party
- B: Bank (Financial Institute for Credit Authority)

(2) Public Key Cryptosystems:

- PKX: Public key of user X.
- SKX: Private key of user X.
- $P_{enc}(PKX, m)$: Encryption of message *m* with public key PKX.
- $P_{dec}(SKX, c)$: Decryption of ciphertext *c* with private key SKX.

(3) Digital Signature Schemes:

- pkx: Verifying key of user X.
- skx: Signing key of user X.
- $< m >_{skx}$: Creation of signature of *m* under signing key skx.
- $S_{veri}(pkx, < m >_{skx}, m)$: Verification of signature $< m >_{skx}$ on message *m, true* for valid and *false* for invalid.

(4) Other items:

- t_x : Timestamp generated by party X.
- $H(m)$: Hash function on message *m*.

3.2 System Setup

There are four parties in our protocol, they are Client, Merchant, TTP and Bank. Client has a pair of public and private keys: PKC and SKC, and a pair of signing and verifying keys: skc and pkc. Merchant has a pair of public and private keys: PKM and SKM and a pair of signing and verifying keys: skm and pkm. TTP has a pair of public and private keys: PKT and SKT. We will employ the technique of proof of equivalence of discrete logarithm to discrete log-logarithm. The above key pairs must follow some overall rule of the whole system. This means that these key pairs must be setup based on the same set of algorithms and parameters. If necessary, the signature scheme of TTP, public key cryptosystem of bank and signature scheme of bank can be defined independently. They need not follow the same set of algorithms and parameters.

At first, we choose three primes to set up the system. The three primes are p, q and q', which are of the form $p = 2q + 1$ and $q = 2q' + 1$. We will use ElGamal cryptosystem for encryption and decryption and a DSA-like scheme for signature.

Public Key Cryptosystems q is the prime number for the ElGamal cryptosystem. Z_q^* is a intractable multiplicative group with order $q - 1$. G is a generator of Z_q^*. SKX is the private key and PKX is the public key. $PKX = G^{SKX}$ mod q and $SKX \in \{1, 2, \ldots, q-2\}$. The ciphertext of m under PKX is:

$$cx = P_{enc}(PKX, m) = (W, V)$$

where $W = G^w \text{mod } q$ and $V = m(PKX)^w \text{mod } q$, w is randomly chosen from $\{1, 2, \ldots, q-2\}$. The message after decryption is:

$$m = V \cdot W^{-SKX} \text{mod } q$$

Digital Signature Scheme p is the prime number for the DSA-like digital signature scheme. Z_p^* is a intractable multiplicative group with order $p-1$. g is a generator of Z_p^*. skx is the signing key and pkx is the verifying key. $pkx = g^{skx}$ mod p and $skx \in \{1, 2, \ldots, q-2\}$. The signature of m under pkx is :

$$< m >_{skx} = (r, s)$$

where $r = g^k \text{mod}\, p$ and $s = k^{-1}(h(m) + r \cdot skx)\text{mod}\, q$. k is randomly chosen from $\{1, 2, \ldots, q-2\}$ and $h(\ldots)$ is the hash function.

For verification of signature, $S_{veri}(pkx, < \text{m} >_{skx}, \text{m})$ is to check

$$r^s \stackrel{?}{=} g^{h(m)} \cdot (pkx)^r \text{mod}\, p$$

Construction of Important Tokens In this section, we will give details of digital tokens used in our fair exchange protocol with credit card based payment.

(1) Credit Card
The token for credit card is of the form

$$\mathcal{C} = \ < C, l, h_1, h_2, \cdots, h_l, E, A >_{skb}$$

The credit token contains the client's identity C, the confidence level l, the expiry date E, maximum credit amount A and $h_i = g_i^x$ mod p, where $g_i \in Z_p^*$ are common generators for $i = 1, 2, \cdots, l$, where x is the concatenation of PIN number, credit card number and salt. The credit token is signed by the bank using its private key skb.

(2) Payment Slip
The data in the payment slip is

$$SlipData = \mathcal{C}, M, O, \$, tc,\ H(\mathcal{C}, M, O, \$, tc),$$

where M is ID of merchant, O is the order, \$ is the amount of money and currency type and t_c is the timestamp generated by the client C.

The payment slip token has the form

$$Slip =< SlipData >_{skc},$$

The payment slip is signed by the client with private key skc.

(3) Encrypted Payment Slip
The encrypted payment slip token is

$$C_S = P_{enc}(PKT, Slip).$$

The client's payment slip is encrypted under the TTP's public key PKT. If necessary, TTP can open it with its private key SKT.

(4) Certificate of Encrypted Payment Slip
$C_S Cert$ is the token to prove C_S is a ciphertext of S without disclosing the signature.

Here, we will give all the details of construction C_S and $C_S Cert.$ p and q are the two prime numbers used in our system. The client has a pair of signing key and verifying key $\{skc, pkc\}$, g is a generator of Z_p^* and $pkc = g^{skc} \bmod p$. The TTP has public key and private key $\{PKT, SKT\}$, G is a generator of Z_q^* and $PKT = G^{SKT} \bmod q$.

For encryption of message m, we have the following:

$$P_{enc}(PKT, m) = (W, V) \bmod q,$$

where $W = G^w$ and $V = m(PKT)^w$, $w \in \{1, 2, \cdots, q-2\}$ is a randomly chosen number.

The signature scheme works as follows: Choose a random $k \in Z_q^*$, the signature has the form

$$Slip =< SlipData >_{skc} \ \equiv \ (r, s)$$

where $r = g^k \bmod p$ and $s = k^{-1}(H(m) + r \times skc) \bmod q$ and $pkc = g^{skc} \bmod p$. $Slip$ is the payment slip.

Encrypting the above payment slip $Slip$ with PKT, we have, $P_{enc}(PKT, Slip) = (W, V)$. The encrypted payment slip with signature is then given as follows:

$$C_S = \{r, W, V\},$$

where $W = G^w \bmod q$, $V = s(PKT)^w \bmod q$.

With transformation $x = G$, $y = W^{-1} \bmod q$, $z = PKT$, $X = r^V \bmod p$, $Y = g^{H(S)}(pkc)^r \bmod p$ and $\alpha = -w$, choose $w_i \in \{1, 2, \cdots, q-2\}$, then

$$t(x_i) = x^{w_i} \bmod q, t(X_i) = X^{z^{w_i}} \bmod p$$

and

$$\begin{array}{l} c = H_l(x||y||z||X||Y||t(x_1)||t(X_1)|| \cdots ||t(x_l)||t(X_l)) \\ c = c_1 c_2 \cdots c_l \\ r_i = w_i - c_i \alpha \bmod q-1 \end{array}$$

(R, c) is the certificate $C_S Cert$ for C_S.

The process of verification is to check,

$$c = H_l((x||y||z||X||Y||u_1||U_1|| \cdots ||u_l||U_l)$$

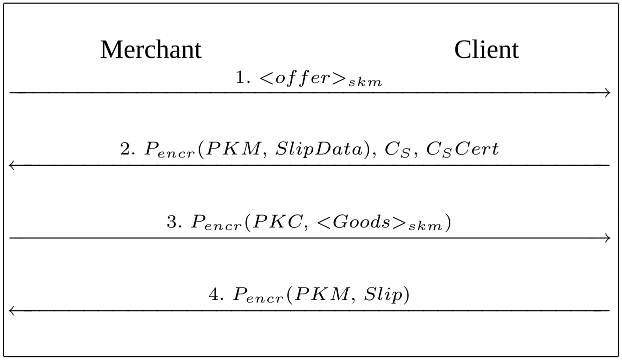
where $u_i = x^{r_i} y^{c_i} \bmod q$, and

$$U_i = \left\{ \begin{array}{l} X^{z^{r_i}} \bmod p \ if \ \ c_i = 0 \\ Y^{z^{r_i}} \bmod p \ if \ \ c_i = 1 \end{array} \right.$$

3.3 Fair Trading Protocol

Based on the tokens defined in the last subsection, our fair trading protocol is constructed. The fairness of the trading between a client and a merchant is guaranteed.

Fair Trading Protocol


Merchant
Client
1. $<offer>_{skm}$
2. $P_{encr}(PKM,\ SlipData),\ C_S,\ C_SCert$
3. $P_{encr}(PKC,\ <Goods>_{skm})$
4. $P_{encr}(PKM,\ Slip)$

For the above protocol, if both the client and the merchant perform properly, the TTP will not be involved. The details of the protocol are as follows:

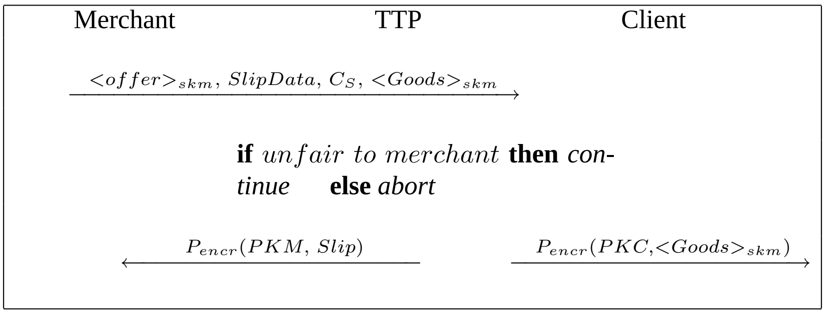
1. In step one, the merchant sends his signed $offer$ to the client. The $offer$ should contain the description of the $Goods$ and related trading information, such as price, valid date etc. The client checks the $offer$, and if client is not satisfied with the $offer$, he can quit the protocol, and therefore it is fair for both parties.
2. In step two, the client sends the merchant his credit card C, order information O, amount of money and currency type $\$$ and time stamp t_c, encrypted payment slip C_S and the certificate C_SCert. The encrypted payment slip C_S is encrypted with TTP's public key. The merchant checks the validity of the above data, and especially, the credit information and encrypted payment slip.

   (1) The merchant checks credit information with equality proof of knowledge (see section 2).

   (2) The merchant uses C_SCert to check C_S is the ciphertext of the payment slip $Slip$ signed by the client (see section 3.2).

   If the merchant finds anything wrong in the above verification, he will quit the protocol, and the protocol will be fair for both parties.
3. In step three, the merchant sends $P_{encr}(PKC, < Goods >_{skm})$ to the client. If the $Goods$ is consistent with the $offer$, the client will continue the protocol. If the $Goods$ is inconsistent with the $offer$, the client quits the protocol. If the merchant believes that it is not fair, he needs to require TTP to run the resolve protocol.
4. In step four, the client sends $P_{encr}(PKM, Slip)$ to the merchant. If the merchant can not get the payment, the merchant will ask TTP to run resolve protocol.

If the merchant can not get the payment, the merchant will ask TTP to run the following resolve protocol:


Merchant
TTP
Client
$<offer>_{skm}$, $SlipData$, C_S, $<Goods>_{skm}$
if $unfair\ to\ merchant$ **then** *continue* **else** *abort*
$P_{encr}(PKM,\ Slip)$
$P_{encr}(PKC,<Goods>_{skm})$

In the completion of the resolve protocol, the merchant has the payment and the client has the goods.

3.4 Properties of Fair Trading Protocol

Some general properties of cryptographic protocols such as integrity and confidentiality are not included in this section, even our fair trading protocol have these properties and can satisfy the related security requirements. Our discussions here only focus on the properties we have emphasized in the design and construction of the fair trading protocol. The fair trading protocol has perfect fairness and high efficiency and provides good availability & reliability of the involved services. The sensitive information (credit card) has untraceability & privacy in the fair trading protocol.

(1) Fairness

If both the merchant and the client behave according to the fair trading protocol, when protocol has completed, client has received the goods and merchant has received the payment. For the client, if something is wrong, he can quit the trading protocol after step three and the whole protocol is fair. For the merchant, if something is wrong after step three, he can bring $offer$, $SlipData$, C_S, $Goods$ to TTP. TTP will check the status. If it is really unfair to merchant, TTP will send the $Goods$ to the client and send the $Slip$ to the merchant. The protocol is fair against cheating attempts by either merchant or client. The protocol is fair in case of system failures as well. The fair trading protocol and the associate resolve protocol can guarantee the trading protocol to be fair in any case.

(2) Efficiency

In normal case, the TTP is off-line and the credit card service from a financial institution is off-line as well. The TTP is only involved when one party misbehaves or system failure happens. The protocol is more efficient than protocols with more on-line components. Computation and communication overheads are reduced to the minimum.

(3) Availability and Reliability

We compare two protocols A and B. If protocol A has one more on-line component than protocol B and all other parts of the two protocols are the same, the on-line component of protocol A has some chance to be unavailable or unreliable because of network problem, system failure or evil behaviors from involved parties or other attackers. Protocol A has less availability and reliability than protocol B. In the fair trading protocol in this paper, TTP and credit card service from a financial institution are off-line in normal case, the protocol is more available and reliable than other protocols with more on-line components (TTP is on-line, the credit card service is on-line or both of them are on-line).

(4) Untraceability and Privacy

The client uses the credit card to pay on the Internet in the fair trading protocol. Untraceability of the credit holder is a necessary or desirable characteristic of this kind of trading protocols. In our fair trading protocol, the credit card is anonymous, the untraceability and privacy of the card holder is achieved.

4 Concluding Remarks

We have introduced our fair trading protocol with off-line anonymous credit card payment over the Internet. The fairness for involved client and merchant is achieved in the protocol and the client is anonymous in the credit card payment. The TTP is off-line and the financial institution for credit service can be off-line as well. The details of digital constructions for credit card payment and fair trading process are provided in this paper. The technique of proof of equivalence of discrete logarithm to discrete log-logarithm is employed as the main building block to construct the protocol. The protocol provides a generic overall solution for fair on-line trading with credit card payment. The involvement of TTP and the on-line financial institution for the credit card service is reduced to the minimum. Our protocol has better efficiency and availability than protocols with more on-line components. The protocol can be used as the starting point to build some complicated protocols in on-line environment, such as on-line gambling protocols.

References

1. F. C. Gartner, H. Pagnia and H. Vogt, "Approaching a formal definition of fairness in electronic commerce", In Proceedings of the International Workshop on Electronic Commerce (WELCOM'99), Lausanne, Switzerland, Oct. 1999.
2. V. Shmatikov and J. C. Mitchell, "Analysis of a fair exchange protocol", Proceedings of the 1999 FLoC Workshop on Formal Methods and Security Protocols, 1999.
3. B.Cox, J.D.Tygar and M.Sirbu, "NetBill security and transaction protocol", Proc. 1st USENIX Workshop on Electronic Commerce, pp. 77-88, 1995.
4. S.Ketchpel, "Transaction Protection for Information Buyers and Sellers", Proceedings of the Dartmouth Institute for Advanced Graduate Studies '95: Electronic Publishing and Information Highway, 1995.
5. A.Bahreman and J.D.Tygar, "Certified electronic mail", Proc. Internet Society Symposium on Network and Distributed Systems Security, pp. 3-19, 1994.

6. R.H.Deng, Li Gong, A.A.Lazar and Weiguo Wang, “Practical protocols for certificated electronic mail”, J.Network and Systems Management, 4(3) pp. 279-297, 1996.
7. N.Asokan, V.Shop and M. Waidner, “Asynchronous protocols for optimistic fair exchange”, Proc. IEEE Symposium on Research in Security and Privacy, pp. 86-99, 1998.
8. R. Hauser, M. Steiner and M. Waidner, “Micro-payments based on iKP”, Technical Report 2791 (No. 89269), June 1996.
9. “SET Secure Electronic Transaction 1.0”, Technical Report, Mastercard, May 1997.
10. M. Sirbu and J. D. Tygar, “Netbill: An Internet commerce system optimized or network delivered services”. ¡http://www.ini.cmu.edu/netbill¿.
11. B. C. Neuman and G. Medvinsky,“Requirements for network payment: The netcheque perspective”, Proceedings of IEEE CompCon’95, March 1995.
12. Y. Mu and V. Varadharajan, “A new scheme of credit based payment for electronic commerce”, the Proceedings of 23rd Local Area Networks Conference, IEEE Computer Society, October, Boston, pp.278-284, 1998.
13. M. Stadler, “Publicly verifiable secret sharing”, Proceeding of Eurocrypto’ 96, LNCS 1070, Springer-Verlag, pp.190-199, 1996.
14. N. Asokan, M. Schunter and M. Waidner, “Optimistic protocols for fair exchange”, Proceedings of 4th ACM Conference on Computer and Communications Security, Zurich, Switzerland, pp.6-17, pp.6-17, 1997.
15. F. Bao, R. Deng and W. Mao, “Efficient and Practical Fair Exchange Protocols with Off-line TTP”, 1998 IEEE Symposium on Security and Privacy, pp.77-85, 1998.
16. J. Camenisch, “Efficient and generalized group signatures”, Advances in cryptology - EUROCRYPT’97, Lecture Notes in Computer Science 1233, Spring-Verlag, Berlin, pp.465-479, 1997.
17. Indrakshi Ray and Indrajit Ray, “An Optimistic Fair Exchange E-commerce Protocol with Automated Dispute Resolution”, EC-Web, pp. 84-93, 2000. http://citeseer.nj.nec.com/462055.html.

Private Reputation Schemes for P2P Systems

Roslan Ismail[1], Colin Boyd[1], Audun Josang[2], and Selwyn Russell[1]

[1] Information Security Research Centre, Queensland University of Technology, Brisbane, QLD 4001 Australia
[2] Distributed Systems Technology Centre, University of Queensland, Brisbane, QLD 4072 Australia

Abstract. The risks from participating in P2P transactions are relatively high. To mitigate such risk a reputation scheme could be applied. Reputation schemes have emerged as a promising means for enabling electronic transactions with strangers. In order to gain optimal results from the reputation scheme, the privacy of feedback provider should be correctly addressed. The feedback provider should be allowed to leave a feedback without fear of retaliation. Unlike in centralized schemes, privacy seems impractical for P2P systems especially when accountability of feedback is also required. This paper considers how privacy can still be provided within the accountability requirement.

1 Introduction

Peer-to-peer (P2P) systems are becoming a popular medium for e-commerce. Intuitively, these systems offer several advantages compared to centralized systems such as being cheaper, more convenient, faster, and also allowing expanded scalability of the systems. Nevertheless, the risks from participating in P2P transactions are relatively high. It is easy to create a phantom transaction and based on such a transaction feedback is given and calculated to produce the reputation rating. Typically this happens because there is no trusted authority to monitor a transaction conducted between a peer and its counterpart.

In contrast, in a centralized system for example, eBay[3], each transaction is monitored by an authority. This measure ensures that the submitted feedback is based on completed transactions. P2P systems commonly have different requirements compared to centralized systems. In the context of reputation schemes, P2P systems require peers themselves to calculate and manage their reputation value on their own. P2P systems may be roughly divided into two categories. The first category is *pure* P2P systems while the second is *mediated* P2P systems. The former operates without involvement of an authority while in the latter the authority participates in certain tasks. In terms of practicality the mediated systems are preferred as it is easy to cheat in the pure P2P systems. In addition, use of unaudited information makes pure P2P unsuitable for formal e-commerce.

It is vital to monitor each transaction that takes place in the P2P systems to prevent false feedback. Recently, Fahrenholtz and Lamersdorf [5] proposed a

[3] http:www.ebay.com

hybrid solution (known as the FL scheme hereafter) which combines centralized and distributed methods to monitor transaction activities. There is an authority known as a *portal* to monitor the feedback process conducted between a peer and its counterparts. Unlike the centralized system, the reputation scheme in the FL scheme is managed by a peer itself. Although the FL scheme seems promising against a false feedback it lacks privacy. No privacy is provided for the feedback provider while submitting feedback. As a result the link between the feedback provider and the submitted feedback is available to the recipient.

Privacy is a vital topic in many electronic systems such as e-voting, e-cash and e-auction. This is equally true for reputation systems. In fact, privacy can help to solve the problem of collecting sufficient negative feedbacks. The feedback providers are usually reluctant to leave negative feedback, even when it is appropriate, for fear of retaliation. Unlike the centralized systems where privacy may not be difficult to implement privacy seems hard for P2P systems.

Another vital property of reputation scheme for P2P systems is accountability. Accountability here means that each feedback should be legitimate. To achieve this property each feedback needs to be signed by the feedback provider. However, by signing the feedback the identity of the feedback provider can easily be traced. Therefore, privacy and accountability seem to contradict one another. This conflict has motivated us to explore a novel way of providing sufficient privacy while at the same time ensuring that accountability of the feedback is not compromised.

In this paper a reputation scheme for P2P systems is applied in which peers calculate and store their reputation on their own without any involvement from an authority. The authority functions as an entity to monitor the process of delivering feedback to appropriate participants. This is vital to ensure privacy is preserved and at the same ensuring feedback is based on a legitimate transaction.

Contribution. The contributions of the paper are twofold: analyzing the security of the Fahrenholtz and Lamersdorf scheme and introducing *privacy* to the scheme.

Organisation of the paper. The remainder of the paper is structured as follows. Section 2 lists related work. Section 3 reviews the Fahrenholtz and Lamersdorf scheme. Section 4 presents our proposal. Finally section 5 discusses several issues and then concludes the paper.

2 Related Work

To understand the implementation of privacy in the reputation scheme, consider a scenario of evaluating the performance of a lecturer in a University. At the end of a semester registered students will be given an evaluation form so that they can leave feedback on the performance of lecturers. To protect the privacy of the students, they are not required to write their names on the evaluation form. By doing so the link between the feedback and students is untraceable. Usually a trusted party ensures that only enrolled students can give feedback and each student can only complete one evaluation form. In the context of e-commerce, the feedback providers are the registered students, the feedback targets are the lecturers and finally the authority refers to the university authority.

Recently, a number of reputation schemes for P2P systems have been proposed for various purposes [3–7]. Gupta et al. [6], for example, proposed a scheme to calculate peers' reputation with the help of an authority known as a centralized reputation calculation agent (CRCA). CRCA is used to maintain consistency in calculating the submitted feedback to produce reputation for peers. In addition, it also can prevent manipulation of the reputation. Since the scheme is aimed at P2P systems the reputation is returned to peers to manage. The scheme is also concerned with protecting the submitted feedbacks from being manipulated by requiring the feedback provider to encrypt and sign the feedback before sending it to CRCA.

Cornelli et al. [3] and Damiani et al. [4] have proposed two schemes (the basic and advanced schemes) to seek a reputable 'servent' (a combination of client and server) for downloading files in P2P systems. Both the schemes employ voting mechanisms to evaluate recommendations collected from others in searching for a reliable servent. An entity receiving the higher vote is a reputable entity. The basic scheme provides partial privacy as it hides the identity of the feedback provider but IP address is in clear form. The IP address is required to verify the vote's origin. The advanced scheme, on the other hand, discloses the identity of voter so that the voter credibility can be assessed. Integrity and non-repudiation of the feedback are assured via encryption and signature mechanisms.

Liau et al. [7] proposed a reputation scheme for P2P systems based on certificate mechanisms. This scheme is a pure P2P reputation scheme; a peer itself is in charge on the management of reputation. This improves the storage and integrity of the reputation rating. The reputation certificate is propagated and evaluated before it can be accepted as a reputation reference. The checking of the reputation certificate is conducted by contacting the recent preceding rater. In a case where the preceding rater is not available the next predecessor rater should be contacted and so on until an available preceding rater is found. To preserve the integrity of the reputation certificate the rater signs the updated certificate.

In all the schemes above the privacy of the feedback provider is not given a fair treatment. Rather the schemes are focussed on how to provide integrity of the feedback. However, we argue that to collect a sufficient amount of feedbacks, especially negative feedback, privacy should be seriously taken into account. Otherwise the problem of eliciting negative feedback remains unsolved. Privacy in fact empowers participants to leave negative feedback when appropriate without fear of possible retaliation. Table 1 presents properties hold by the reviewed schemes. The • denotes a full feature is available while ⋆ denotes a partial feature is provided. Out of three schemes reviewed, we choose to improve the scheme due to Fahrenholtz and Lamersdorf because it has two suitable features for P2P systems. The first feature is a monitor mechanism to check that transactions

Table 1. Summary of properties in several reputation schemes

	Integrity	Non-Repudiation	Privacy	Monitoring
Liau et al. scheme	•	•		
Cornelli et al. scheme	•	•	⋆	
Damiani et al. scheme	•	•	⋆	
Gupta et al. scheme	•	•		
Fahrenholtz and Lamersdorf scheme	•	•		•

have been conducted by the peers. By doing so a feedback is assured to be based on a legitimate transaction. The second feature is a mechanism which prevents peers from discarding unfavorable feedback collected by them.

3 Fahrenholtz and Lamersdorf Scheme

Fahrenholtz and Lamersdorf [5] proposed a distributed reputation scheme for P2P networks. In the FL scheme an entity called a portal is used to monitor transactions conducted between peers. The portal also records the number of transactions conducted, as well as the number of feedbacks obtained by each peer. This is achieved via the use of ticket and nonce. The ticket typically contains identification of a peer and its counterpart, and the nonce. The nonce is extracted from the transaction ticket and it needs to be submitted alone with the questionnaire form. These mechanisms can prevent peers from discarding unfavorable feedbacks collected. To achieve integrity and non-repudiation of the feedback each one will be encrypted and signed before sending it one to another.

3.1 Outline of FL Scheme

Figure 1 shows the entities in the scheme and their interactions. There are two types of entities in the scheme; a trusted third party (TTP) known as a portal and the peers. The peers are required to register with the portal before commencing with a transaction. Each peer is required to create a key pair (private, public) when completing the registration. The scheme contains five phases; authentication of reputation management system subjects, service location for a context-specific transaction partner, selection of transaction partner, domain dependent transaction and rating partner. For simplicity we only review phase 2 and 5 (for further details consult [5]).

- **System Setup.** Let U_1 and U_2 be two peers who want to transact one to another, and A be an authority. The identities of U_1 and U_2 are denoted by

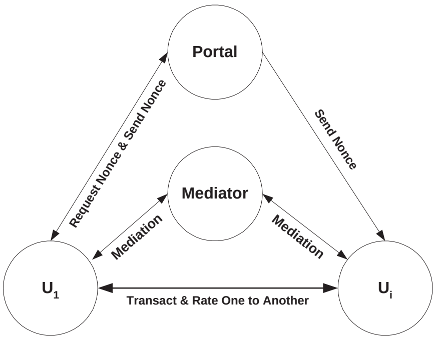


Fig. 1. Abstract View of FL scheme

id_{U_1} and id_{U_2}, respectively. $S_X(m)$ denotes that m is signed by X, $E_{Y,Z}(m)$ denotes m is encrypted using a shared key between Y and Z, and transaction tickets for U_1 and U_2 are denoted by d_{A,U_1}=$(id_{U_1}, id_{U_2}, r_{A,U_2})$ and d_{A,U_2}=$(id_{U_1}, id_{U_2}, r_{A,U_1})$, respectively. The notations r_{A,U_1} and r_{A,U_2} represent the nonce issued by an authority A for user U_1 and U_2, respectively.

- **Transaction Partner Selection.** Identification of a suitable partner must take place before a transaction begins and it is assumed that this process has already taken place beforehand. Rather we continue to the next step where U_1 requests the authority A to issue transaction tickets to itself and its counterpart. Upon receiving this request, two tickets d_{A,U_1}, d_{A,U_2} are issued and signed by A where d_{A,U_1} is for the transaction ticket for U_1 and d_{A,U_2} is for U_2. To ensure confidentiality and integrity of these tickets, they are encrypted with the key shared between A and U_1, and A and U_2, respectively. The following are the protocol messages sent by A.

  1. $A \to U_1 : E_{A,U_1}(d_{A,U_1}, S_A(d_{A,U_1}))$
  2. $A \to U_2 : E_{A,U_2}(d_{A,U_2}, S_A(d_{A,U_2}))$

- **Rating of Partner.** Upon completing the transaction, U_1 and U_2 can start to rate the performance one to another. U_1 sends the $Q'naire_{Cxt,U_1}$ along with the nonce to U_2, and vice versa. Upon receiving the $Q'naire_{Cxt,A}$, U_1 and U_2 can start to fill it with a feedback before sending it to one another. To protect the integrity of the $Q'naire$, it is encrypted with the key shared between U_1 and U_2. To confirm that the completed $Q'naire$ has already been submitted to one another U_1 and U_2 send the nonce to A. The nonce can be extracted from the transaction ticket. Upon receiving the nonces A sends an acknowledgement to both U_1 and U_2 to indicate the status of the sending nonce: either it is fine or an error is reported. The nonce functions as a means to prevent U_1 and U_2 from discarding unfavorable feedbacks. The protocol executed is described as follows.

  1. $U_1 \to U_2 : E_{U_1,U_2}(Q'naire_{Cxt}, S_{U_1}(Q'naire_{Cxt,U_2}), r_{A,U_2})$

2. $U_2 \rightarrow U_1 : E_{U_1,U_2}(Q'naire_{Cxt,U_1}, S_{U_2}(Q'naire_{Cxt,U_1}), r_{A,U_1})$
3. $U_1 \rightarrow A : E_{A,U_1}(r_{A,U_1})$
4. $U_2 \rightarrow A : E_{A,U_2}(r_{A,U_2})$

3.2 Analysis of the FL Scheme

Notation. Let N_U be the number of transactions carried out by user, F_A denotes the number of nonces received by the authority, F_U denotes the number of feedbacks obtained and recorded by the user.

The behavior of the relying party depends on the relative sizes of N_U, F_A and F_U. We consider several different cases.

1. $\mathbf{N_U = F_A = F_U}$.
   This outcome means that the number of transactions made by a user U is equal to the number of feedbacks for the user recorded by the authority and the user. This is an ideal case where all the participants follow the protocol honestly. Typically in this case the reputation rating of the user is accepted.
2. $\mathbf{F_U = F_A}$ and $\mathbf{F_U < N_U}$.
   This outcome means that the number of feedbacks recorded by user U is equal to the number of feedback recorded by authority but it is less than the number of transactions recorded by the authority. There are occasions when some feedback providers may not return their feedback after completing the transactions. This could be quite common especially when there is no reward for submitting a feedback. In addition, fear of the consequences due to the given feedback is another factor which causes lack of interest to leave a feedback. In this case the reputation of the peer is commonly accepted.
3. $\mathbf{F_U > F_A}$ and $\mathbf{F_A = N_U}$
   This outcome means that the number of feedbacks recorded by user U is greater than the number of feedbacks and transactions made and recorded by the authority. The user may create some phantom feedbacks in order to boost his reputation. Thus, in this case the reputation of the user is not accepted.
4. $\mathbf{F_U < F_A}$ and $\mathbf{F_A = N_U}$
   This outcome means that the number of feedbacks recorded by a user U is less than the number of feedbacks and transactions made recorded by the authority. The user may have discarded some of unfavorable feedbacks submitted for him. The reputation of the peer is not accepted.
5. $\mathbf{F_U > F_A}$ and $\mathbf{F_U < N_U}$
   This outcome means that the number of feedbacks recorded by a user U is greater than the number of feedbacks recorded by the authority but less than the number of transactions made. Some feedback providers may choose not to return their nonces to A. As a result A's record on the number of nonce received will be less than the number of feedback record by the user. The reputation of the user could be accepted depending on a proof provided by the user.

Although the FL scheme provides sufficient protection from the outsider attacks it is exposed to attacks by the internal players. For simplicity the communication line is assumed reliable. The nonce is always reached to the authority. In the FL scheme the portal is a trusted party. However, this may not be valid in some cases. We would like to point out some important facts regarding the above outcomes. For example, in case 2 an attack could be launched by the dishonest portal. The portal may learn the number of transactions conducted by U_1, as well as the number of feedbacks collected by U_1 from the returned nonce. With this knowledge a phantom nonce can be added to U's account. As a result U_1's reputation could be rejected as there is discrepancy in the record of the authority and U_1.

A major restriction the FL scheme possesses is that the feedback target may learn the link between the feedback provider and the feedback because it is in clear form. As a result the feedback providers may be reluctant to leave honest feedback especially in the case of negative feedback due to the consequences they may suffer later. In a real e-commerce environment negative feedback is needed to counterbalance positive feedback so that the produced reputation can reflect true behavior of users.

4 Improved Scheme

We propose an improvement of the FL scheme by introducing privacy as the major concern. The players and processes are similar to the FL scheme. To ensure that privacy can be achieved the number of a peer's counterparts should be more than one, and preferably a large number. This requirement is essential to allow unlinkability. The importance of unlinkability to maintain privacy is discussed by Maitland et al. [8]. Without sufficient number of players privacy seems impossible to implement. Another vital consideration is the timing of delivering of the feedback. If the feedback is sent immediately after the transaction is completed then the link between the feedback and the feedback provider can easily be formed. To avoid such undesirable outcome delay of the delivery of the feedback to a certain time later could be undertaken. This could either be a randomised delay, or delivery of feedbacks could be batched after a threshold number of feedbacks has been received.

Unlike the FL scheme, our proposal does not use shared keys. Instead the peers use their counterpart's public key to encrypt the feedback. However, we follow the practice of FL scheme in managing the administrative task. The authority is still responsible for issuing a nonce to each peer. The nonce must to be obtained before a transaction can take place. Besides issuing the nonce, the authority also maintains a record of the returned nonces.

There are six phases in our proposal; requesting nonce, preparing and sending token, signing and sending legitimate token, submitting feedback, calculating feedback and showing reputation. In our improved scheme, registration of participants is not required. Thus, each participant is assumed to have a valid certificate issued by the certificate authority CA.

There are several options can be taken to construct our improved scheme. One can construct a scheme based on ring signature schemes [9]. Ring signatures allow the identity of the signer to be hidden from recipients while retaining the important advantage of enabling accountability of feedback to be achieved. Thus the two characteristics we require for reputation schemes are provided. However, this option does not seem very practical to be implemented as it requires vastly more computation to verify the signatures. There are two types of computation required. The first is to verify signature of the n members in a ring signature for a feedback provider. The second is to verify each of the d feedback providers. As a result there are $d \times n$ computations are required. A second option is to implement a scheme based on a bilinear ring signature scheme. This scheme reduces the inefficiency faced by the first option but like the first scheme it is still too inefficient to be really practical. To overcome this inefficiency, a token based scheme is proposed. Since our scheme follows the same process as the FL scheme where a nonce is required to be obtained before starting a transaction this phase is not considered in the following protocol.

4.1 Protocol of the Scheme

The following protocol uses several notations as follows: ID_{FP} denotes an identification of the feedback provider, E_X denotes encryption using X's public key and Sig_X denotes signing using X's private key.

Preparing and Sending Token. FP prepares the transaction particulars m. The content of m can be the date of transaction, the feedback target's identification, the amount of transactions and the given feedback. To ensure integrity is achieved a pair (ID_{FP}, m) is encrypted using TTP's public key before sending it to TTP for signing. In a variant of this procedure the token could be created using electronic cash technology [2]. A coin is issued by the TTP for a particular transaction and when submitting feedback the coin payment protocol is used. The advantage of this option is that the feedback value can be hidden from the TTP, but there is an extra computational cost. We do not consider this option further in this paper.

$FP \rightarrow TTP : E_{TTP}(ID_{FP}, m)$

Signing and Sending Legitimate Token. Upon receiving the pair (ID_{FP}, m) from FP, TTP decrypts it and then verifies the correctness of m against a database of transactions maintained by TTP itself. If the verification is successful m is signed by TTP and then encrypted using the FP's public key. To complete the phase, TTP sends m to the feedback provider. The signed m is considered a legitimate token. Only the legitimate token can be used for submitting a feedback. Without using the legitimate token the feedback will not be counted for calculation of reputation. A nonce is also issued by TTP and then submitted to FP.

$TTP \rightarrow FP : E_{FP}(Sig_{TTP}(m), nonce)$

Submitting Feedback. FP sends to FT the legitimate token m which consists of the feedback. To protect integrity of the legitimate token, m is encrypted using the feedback target's public key. FP also send the nonce to FT.

$FP \rightarrow FT : E_{FT}(Sig_{TTP}(m), nonce)$

Calculating Feedback. Upon receiving m, FT decrypts and then verifies the TTP's signature on m. If the verification is successful the legitimate token is accepted otherwise it is rejected. The accepted token is then used to calculate the reputation of the feedback target. FT sends the nonce to TTP to confirm the feedback is received from the feedback provider. However, FT does not who is the feedback provider.

TTP sends a signed list n to the feedback target. The list n consists of number of the legitimate tokens issued by TTP for the feedback target. n is important to prevent the feedback target from discarding the submitted legitimate tokens. In addition, n also acts a means to convince the relying party that the calculated reputation is based on the submitted feedback.

$A \rightarrow FT : E_{FT}(Sig_{TTP}(n))$

Showing Reputation. Before a transaction can commence, FT sends the calculate reputation and n to the relying party so that the relying party can evaluate the validity of reputation. Due to a possibility of having huge number of tokens to be verified the relying party could batch them. The scheme proposed by Bellare et al. [1] can be employed which save the computation of verification. The relying party has to verify two signatures: the TTP's signature on the tokens and the TTP's signature on the list n.

5 Analysis

Privacy. The improved scheme achieves conditional privacy where the feedback provider is hidden from public except TTP. However, it requires trust to be placed on the trusted third party not to reveal the identity of the feedback provider otherwise the privacy of the feedback provider is compromised. In other words TTP is assumed honest in performing its task. However, in a case where this assumption is difficult to implement, for example, in the presence of the dishonest TTP then threshold schemes could be implemented. This means a number of TTP is required in which each individual TTP shares a portion of identity of the feedback provider. Without sufficient number of TTP to form the identity of the feedback provider, the privacy of the feedback provider is preserved.

6 Conclusion

An analysis conducted on the scheme of Fahrenholtz and Lamersdorf reveals a few security concerns. We have proposed an improved reputation scheme which is based on the FL scheme. The improved scheme provides privacy for feedback providers while submitting a feedback. With this property the feedback providers can leave negative feedback without fear of retaliation from the other parties. The token based solution is suitable to provide a simple privacy protection and furthermore it is efficient in terms of computation required to verify signatures.

References

1. Mihir Bellare, Juan A. Garay, and Tal Rabin. Fast batch verification for modular exponentiation and digital signatures. In K. Nyberg, editor, *EUROCRYPT '98*, volume LNCS 1403, pages 236–250. Springer-Verlag Heidelberg, 1998.
2. Stefan Brands. Untraceable off-line cash in wallets with observers, In D. R. Stinson, editor, *Advances in Cryptology - Crypto '93*, pages 302–318, Springer-Verlag, 1993.
3. Fabrizio Cornelli, Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. Choosing Reputable Servents in a P2P Network. In *Proceedings of the eleventh international conference on World Wide Web*, pages 376–386. ACM Press, 2002.
4. Ernesto Damiani, Sabrine De Capitani di Vimercati, Stefano Paraboschi, Pierangela Samarati, and Fabio Violante. A reputation-based approach for choosing reliable resources in Peer-to-Peer networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 207–216. ACM Press, 2002.
5. D. Fahrenholtz and W. Lamersdorf. Transactional security for a distributed reputation management system. In K. Bauknecht, A. Min Tjoa, and G. Quirchmayr, editors, *Proceedings of the 3rd International Conference on Electronic Commerce and Web Technologies*, volume 2455 of *Lecture Notes in Computer Science*, pages 214–223. Springer-Verlag, 8 2002.
6. Minaxi Gupta, Paul Judge, and Mostafa Ammar. A reputation system for Peer-to-Peer networks. In *ACM 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2003)*, pages 144–152. ACM Press, June 1-3 2003.
7. Chu Yee Liau, Xuan Zhou, Stephane Bressan, and Kian-Lee Tan. Efficient distributed reputation scheme for Peer-to-Peer systems. In *The 2nd International Human.Society@Internet Conference*, volume LNCS 2713, pages 54–63. Springer-Verlag, 2003.
8. Greg Maitland, Jason Reid, Ernest Foo, Colin Boyd, and Ed Dawson. Linkability in Practical Electronic Cash Design. In *Proceedings of Information Security Workshop (ISW 2000)*, volume LNCS 1975, pages 149–163. Springer-Verlag, 2000.
9. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In C. Boyd, editor, *ASIACRYPT 2001*, volume LNCS 2248, pages 552–565. Springer-Verlag, 2001.

Filtering spam at e-mail server level with improved CRM114

Víctor Méndez, Julio Cesar Hernandez, Jesus Carretero, Felix García

Department of Computer Science
Universidad Carlos III de Madrid

vmendez@inf.uc3m.es, jcesar@inf.uc3m.es,
jcarrete@inf.uc3m.es, fgcarbal@inf.uc3m.es

Abstract. Security managers and network engineers are increasingly required to implant corporative spam-filtering services. End-users don't want to interact with spam-classify applications, so network engineers usually have to implement and manage the spam-filtering system at the e-mail server. Due to the processing speeds needed to put these solutions into work at the server level, the options at hand are reduced to applications of the black-list/white-list type. This is the reason behind the fact that most applications based on AI techniques run only on the client side, particularly those based in the Naïve Bayes scheme, which has proved to be one of the most successful approaches to fight against spam, but nowadays is not as fast as other techniques and still not able to process the high amount of email traffic expected at a mail server. However, spam mutates and the spamies techniques have quickly evolved to easily pass the traditional black/white list applications, so there is a compelling need for the use of more advanced techniques at the server level, notably those based in the Naïve Bayes algorithm. This article explores this possibility and concludes that, simple improvements to a well-known Naïve-Bayes technique (CRM114[2]), following some ideas suggested in [8], could turn this algorithm into a much faster and significantly better one that, due to these improvements in speed, could be used at the server level.

1 Introduction

The problem of automatically filtering unwanted email messages is one of increasing importance, since bulk emailers take advantage of the great popularity of the electronic mail communication channel for indiscriminately flooding email accounts with unwanted advertisements. There are many factors which contribute to the proliferation of unsolicited spam, specially the inexpensive cost of sending email[14], and of obtaining pseudonyms[15]. On the other hand, we have the high cost associated with users receiving spam[2] and the network overflow.

The spam filtering problem can be seen as a particularly instance of a Text Categorization problem where the classes are Spam or Ham. In recent years, a vast amount of techniques have been applied to solve this problem. Some of the top-performing methods are Learning Rules that Classify e-mail[20](1996) based in the RIPPER algorithm, Ensembles of Decision Trees[16](1999), Support Vector Machines[17](1998), Boosting[18](2000), and Instance Based Learning[19](2000). Nowadays, advanced Naive Bayes methods are the top-performing ones, coming from Paul Graham principles for spam-classifying [3], and some basic improvements [4](2003). The false positives go from 0,3% to 0,06%, and the detected spam from 99,5% to 99,75%[4].

On this paper we are going to explain principles that makes CRM114 one of the best accuracy filtering application. We after compare the design features and obtained results with other state-of-the-art applications, and expose our approach to the problem modifying CRM114 behavior on window size for empirically comparing accuracy versus speed, and on features text extraction, introducing the concept of virtual feature, that will finally modify the original SBPH polinomy used on CRM114. We present two experiments results and we extract some conclusions.

2 Sparse Binary Polynomial Hashing (SBPH) CRM114

SBPH creates a lot of distinctive features from an incoming text, then the Naive Bayes technique is applied to this features instead of directly to the words. For this purpose the algorithm slides a window of length five words over the incoming text and, for each window state, generates a set of order-preserving sub-phrases containing combinations of these words. These order-preserved sub-phrases are processed calculating 32-bit hashes, and with all the resulting sub-phrases the algorithm creates a 32-bit superhash (i.e. hashing of hashes) value that will be used to calculate the Naive Bayes probabilities. Essentially, each sub-phrase tries to extract a word feature from the text. With a window size of five words, each word affects 2^{5-1}=16 features.

The performance of CRM114 Mailfilter from Nov 1 to Dec 1, 2002: 0.068% of false negatives and ZERO false positives[2]. Its filtering speed on classification is less than 20Kbytes per second (on a Transmeta 666 MHz) [2], which obviously hinders the use of crm114 for filtering at the mail-server. For this purpose, an average network needs a classification speed of at least 60kb/sg [13].

3 CRM-114 and other state of the art filtering applications

Every application has distinctive characteristics that we summarize on the next diagram, where we show the features for some state of the art antispam applications.

Table 1.

	1	*2*	*3*	*4*	*5*	*6*	*7*	*False positives*
Crm114 [2]	Yes	5	S	Yes	GPL	M, S (?)	crm	0.00% [2]
SpamAssesine [6]	Yes	Unknown	A	Yes	No	M, I	Unknown	0.19% [6]
Gnus-emacs [7]	Unknown	No	N	No	GPL	M(+IMAP)	gnus	Unknown
SpamProbe [8]	Yes	1,2	S	Yes	QPL	P(+fetchmail)	No	0.035% [8]
PopFile [9]	Yes	Unknown	A	Yes	GPL	M, P.	PERL	0.125% [10]

1.Automatic featured extractor from text
2.Phrase window size.
3.HTML filter: No HTML filter (N), Simple HTML filter (S), Advanced HTML filter process (A)
4.Specific design to arise with false-positives.
5.Software license.
6.Filtering level at the: MTA Client (M), POP3 Proxy (P), At the e-mail server (S) Internet level [6](I).
7.Generic filter language [2].

All the applications are based on the naive-bayes algorithm, except SpamAssassin, which is based on a combination of a GA and rules, which is probably the reason of the worse false positives rate. The false positives rates are taken after different learn cycles, depending on the application approach to the optimum value. We can see this data as a kind of "how good can it do it". The classifying experiments were made with a different number of mails; every author has used different sets, but always in the order of thousands. It is clear from the table above that crm114 has the better false positive rate.

Automatic featured extractor for text is very important for the detection of new spamies techniques [5], which simplify the network engineer task of coping with new tactics. It is also known that both crm114 and SpamProbe use a similar windowed word philosophy, which will be explained below.

All the applications except gnus-emacs use some type of HTML processing. This is becoming important due to the fact that a lot of spamies techniques are based on HTML use [9]. We can also see that gnus-emacs has not implicit design to manage false positives.

Additionally, some kind of free software license is needed to give the network engineer the possibility of escalating or updating the code, or for tuning in a production domain without the strategic dependency on a specific software developer. If not a completely open code license, at least a specific generic filter language that allows for some level of implementation-specific tuning should be offered. This is especially important on spam-classify applications because they generally don't have good generalization features but are able to successfully operate in a real world domain after only small design changes. From our point of view, the main drawback of SpamAssassin for our purpose is that it has not free software license or an open generic filter language, so it works outsourcing the spam-filter at internet level, a solution that is not appropriate for a corporative implementation at the server level. On the other hand, crm114 could filter at both the MTA Client level and also at the server level, but only if we could greatly increase its classification speed. The rest of applications could run only on the client side.

4 The window philosophy

The crm114 algorithm uses a window size of five words. Most researches like Brian Burton [8] indicate that window sizes over two words generally produce no better accuracy or false positives rates, and in fact may well be worse because they could lead to an overflow of features and, additionally, they greatly decrease the filtering speed. It is obvious that crm114, which is a combination of an advanced Naive Bayes method and polynomial hashing, has a very different window philosophy: bigger windows gives polynomial techniques a more relevant weight in the final combined method. If we set the window size to two words, we will use a two variables polinomy that is less suitable than a bigger polinomy for feature extraction. The first consideration we have to do is that following the crm114 principles for relationship between window size and features, for a two words windows size we may extract only 2 features, and this sounded a little too poor. So at the end of the day, we think in trying different empirical experiments playing with both window size and word features. For this purpose we converted the static crm114 compiler into a dynamic matrix of pipelines (window size) and superhashes phrases (number of words features) to help on false positives service level and classify rate decisions.

5 The virtual features

How are we going to extract different number of features than SBHP features relationship with the window size, in order of 2^{N-1} **?** We are not going to follow such relationship, for example with 2 words window size we repeat original SBHP coefficients patterns, $2^{2-1} = 2$ patterns until 20, and we call them virtual features.
Depending on the conjunction of window size and virtual features, we will obtain diferents SBHP functions, taken as algorithmical seed the original Yerazunis function with 5 x 16 dimension.

6 Test I

6.1 Benchmark corpus

Our benchmark corpus contains the learning mails set to create the .css files, (superhashes mapped files). Yerazunis recommends a learning corpus around 0,5 Mb size, and following his recommendation we have used the file nonspamtext.txt (695111 bytes extracted from my personal inbox and public mail lists asfsdevel, SI-edu, or SL-admin) and the file spamtext.txt (536547 bytes from a public set [11]).

The classify mails set to test our approach come from individuals donations [12], thus the learn/classify sources are independent enough to generalize results at the mail server filter level. Following Paul Graham indications, they approximately have the same ham/spam distribution (ham=170, spam=220)

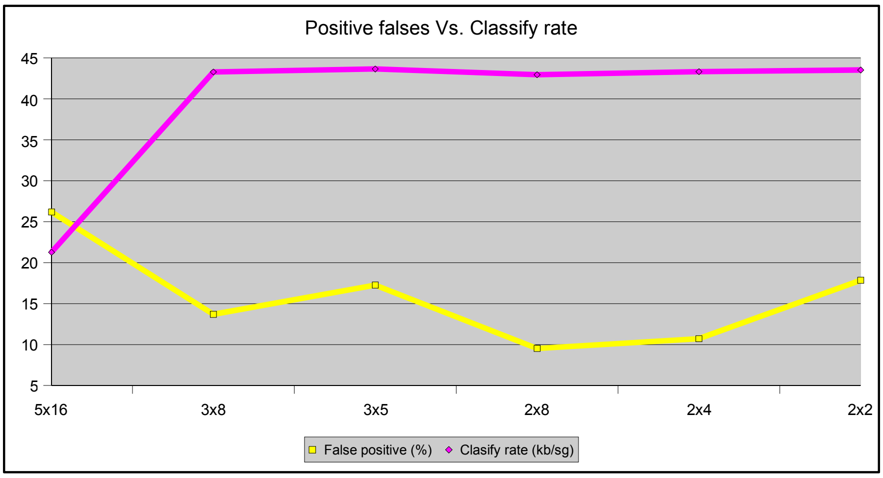
6.2 Results

The tests show the result for single-pass learning, without any retraining cycle, so results do not shown a "how good can it do it", but are enough for comparison between different matrix sizes. The test were done on an 700Mhz. Intel Pentium III Copermine (c) with 128Mb memory, and a processor load over 99% for the classify process. The times are taken strictly on classify process part. The matrix size are relative to pipelines(window size) x superhashes phrases.

Table 2.

	5x12	3x8	3x5	2x8	2x4	2x2
Spam detected	87,61%	88,53%	94,95%	92,20%	92,66%	95,87%
False positive	26,19%	13,69%	17,26%	9,52%	10,71%	17,86%
sg. training spam	496,67	376,86	374,05	373,34	373,6	374,98
Sg training ham	36,91	27,99	26,37	25,82	26,27	25,75
Training rate kb/sg	2,25	2,97	3	3,01	3,01	3
Classify spam	50,48	24,33	24,32	24,32	24,52	24,41
Classify ham	62,61	31,25	30,81	31,7	31,01	30,87
Classify rate kb/sg	21,27	43,28	43,64	42,95	43,32	43,53

Fig. 1.



The graph above show the values for the critical parameters. This confirms worse false positive rates if the window size is over two words. Remember the original static crm114 matrix size is 5x16, so crm114 has similar behavior on the window size value to the Brian Burton study for the SpamProbe[8], which obtains betters results on 1 and 2 words window sizes. We also obtain much better classify speed rates but this was an obvious result because, for example, the original crm114 has to calculate 5x16=80 hash for a superhash feature, and, at the best false positive performance only 2x8=16 hash are to be computed for a superhash feature.

6.3 Test I Conclusions

We have proposed a new approach for implementing spam filtering on the email server which is a modification and also an improvement over the state-of-the-art crm114 technique and leads to much higher speeds, due to the fact that it uses a window word size of only two words and, surprisingly enough, also to better classification and false positive rates.

7 Test II

Now we would like see how our approach works with a bigger test corpus, in order of thousands mails from SpamAssasine public corpus[21]. We also play with relearning cycles, following our aim of ZERO false positives, that original crm114 may accurate[2], and checking if our conclusions for a one cycle of Test I are also valid in a real domain making maps proccess. For this purpose we first train up to medium size corpus, we after relearn every false classificated case up to final corpus size, with bigger ham corpus than spam one, trying to force more ham weight on maps, for better results on false positives. We finally test from a different corpus for statatistics. We will combine in a natural way the .css maps and the test corpus, for example easy ham map and spam map, testing with easy ham and spam corpus(EASY-EASY); and we also check the cross map-corpus tests, for example easy maps with hard test corpus, that are not expected to get good results, but we want to check it.

7.1 Benchmark corpus II

We focus our study on 2 words window size and original crn114 5x16 matrix for comparison.

On the first two phases we take mails from 2003 SpamAssasine public corpus, which has a singular ham sources classification with easy to classify ham, and hard ham that usually produces a worse false positives rate, so we are going to test with a ham map done with easy ham, and other with a mix of easy and hard ham on first phase, and only hard ham on relearn phase, we will call it mix-ham, but is some kind of hard ham with little enough easy ham. After relearn classifying thousands mails, we obtain map files of the following corpus size and mails number:

Table 3.

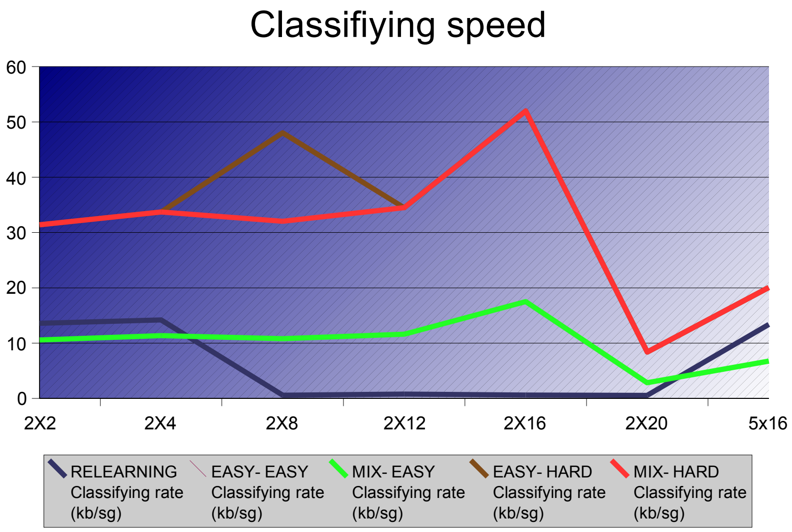
	2x2		*2x4*		*2x8*		*2x12*		*2x16*		*5x16*	
Mail class	***Bytes***	***Mails***	***Bytes***	***Mails***	***Bytes***	***Mails***	***Bytes***	***Mails***	***Bytes***	***Mails***	***Bytes***	***mails***
spam	349107	67	349317	65	331767	60	314745	54	332300	50	308988	67
Easy-ham	508550	62	500116	62	498545	62	500256	62	405325	63	292499	60
mix-ham	414676	48	419989	46	677018	46	673904	44	653895	41	416095	53

Training corpus:	Spam Assasin public corpus 2002. Mails:
Spam:	501
Easy ham:	2551
Hard ham:	250

7.2 Test II Results

The test were done on an 700Mhz. Intel Pentium III Copermine (c) with 128Mb memory, and a processor load over 99% for the classify process. The times are taken strictly on classify process part. The matrix size are relative to pipelines(window size) x superhashes phrases.

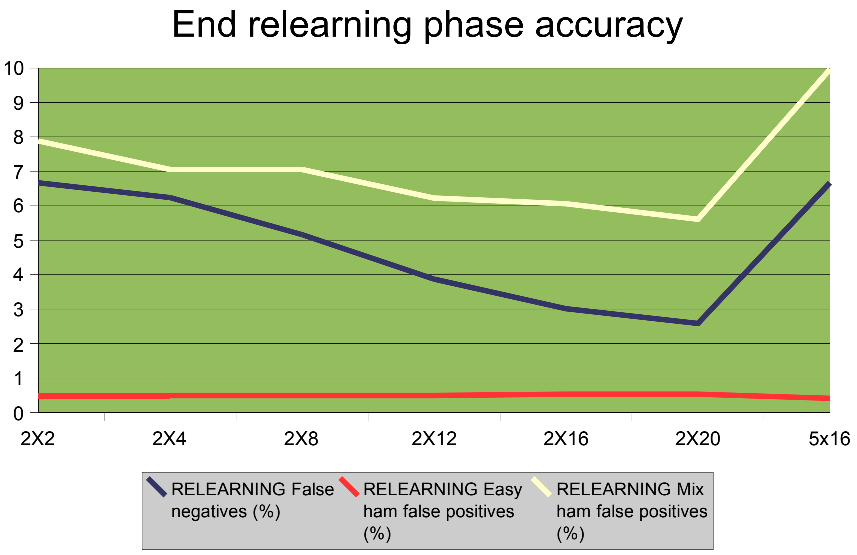
Fig. 2.



The classify speed shown on the graph above, confirm similar conclusions than Test I. The relearning classify rate has better results on original 5x16 matrix, than some of the two words window size matrix. But this data are not relevant because the critical classify speed is on production time, not in relearning phase. On the other hand we can see the testing classify rates, working better with 2 words window size, specially with 12 and 16 features, and with the bests results on tests that uses hard ham corpus. We also can see better results on original 5x16 matrix than in the 2x20, so 2x20 will be out of consideration because speed deficiencies.

For the next graph we have to do the consideration that during relearning phase the maps are changing, training the maps with every false classificated case of the test. So the accuracy will also change and the data we shown are taken from the beginning to the end of relearning phase, for comparison with the test phase accuracy(see below).

Fig. 3.

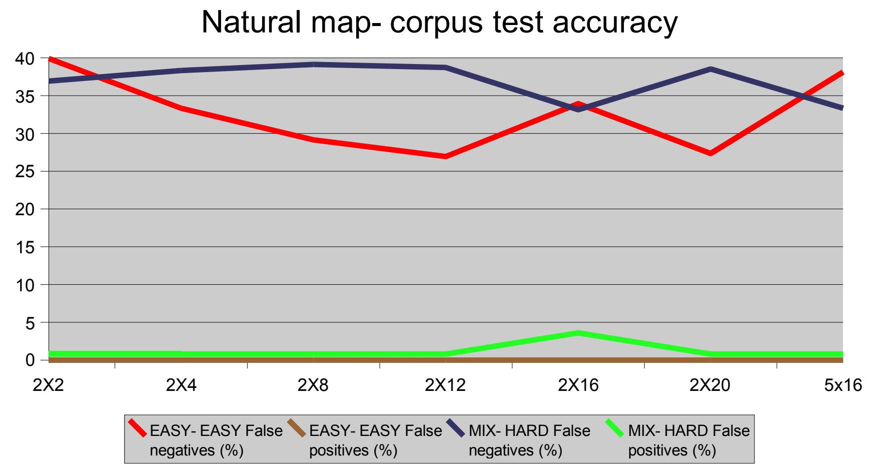


We confirm better results on two words window size, and much better with more features. We also confirm the easy and hard ham SpamAssasine classification, that goes for false positives from around 0,5 % value for easy ham, to more than 6% with hard ham.

Diagram below shown the accuracy for natural tests: one is the spam and easy ham maps versus spam and easy ham test corpus, the other is spam and mix ham maps versus spam and hard ham corpus.

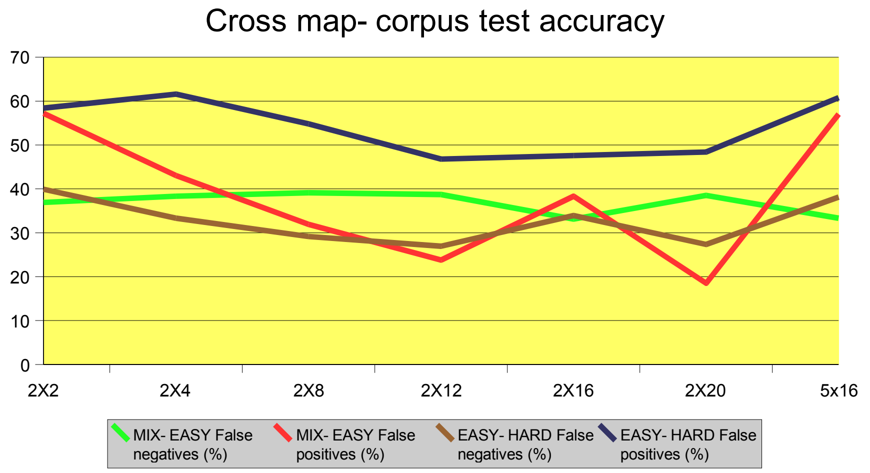
We get the ZERO false positives for easy ham case. The best results are in 2x12 and 2x20 but we have to remember that 2x20 has the big problem of speed. The green and brown lines are the false positives and our previous working thesis of "more weight on ham maps for better false positives results" obtain here empirical confirmation, if we compare with relearning accuracy, where the maps at beginning were of similar size. But in the other hand we have very bad results on false negatives. We can clearly see this dependency in the 2x16, where mix-hard false negatives decrease if we compare with other matrix size, but false positives increases. The important fact for our study is that we can diminish false positives playing with learning and relearning final mails corpus size, but increasing the false negatives. We may tuning for a agreement solution to obtain also ZERO false positives with not so bad false negatives.

Fig. 4.



The next diagram shown that not natural map-corpus combination are not good to work.

Fig. 5.



Conclusions

After the more reliable Test II, we confirm the conclusions of Test I, our approach has better accuracy and speed than original 5x16 static matrix size crm114. We also extract more conclusions, the most important is that the ZERO false positives are within reach if we use relearning on false cases in a planned way, and we have proposed a valid tactic in two phases for this aim. An other conclusion is we can give more important to the market critical parameter "false positives", using bigger ham corpus size than the spam one, to make the maps (.css); but this decrease the other false cases, the negative, the spam and ham maps (and they train corpus) are an antinomy with absolute dependencies one to each other for the final results. We also observed that a ham subdivision in hard an easy may be good for planing the train and relearning tactic, we have to considerer that hard ham is not very used, but are those mails that could be easy mistaken with spam, even for the human eye, so depending on corporative domain we should specially train the maps for them, or not. Finally we have shown that not natural map-working domain, are not valid at all.

References

[1] Tom M. Mitchell. Machine Learning - McGraw-Hill, ISBN: 0-07-042807-7

[2] William S. Yerazunis. Sparse Binary Polynomial Hashing and the CRM114 Discriminator - MER Labs. Cambridge, MA. 2003 and Cambridge Spam Conference Proceeding - http://crm114.sourceforge.net/

[3] Paul Graham. A Plan for Spam. 2003 Cambridge Spam Conference Proceeding http://paulgraham.com/spam.html

[4] Paul Graham. Better Bayesian Filtering. 2003 Cambridge Spam Conference Proceeding http://paulgraham.com/better.html

[5] Jason D.M. Rennie, y Tommie Jaakkola. Automatic Featured Induction for Text Classification. - MIT, AI Labs. Abstract Book. 2002 and 2003 Spam Conference- http://www.ai.mit.edu/~jrennie/spamconference/

[6] Matt Sergeant. Internet Level Spam Detection and SpamAssassin 2.50.- 2003 Cambridge Spam Conference Proceeding - http://axkit.org/docs/presentations/spam/

[7] Teodor Zlatanov. Spam Analisys in Gnus with spam. - 2003 Cambridge Spam Conference Proceeding - http://lifelogs.com/spam/spam.html

[8] Brian Burton. SpamProbe: Bayesian Spam Filtering Tweaks - 2003 Cambridge Spam Conference Proceeding - http://spamprobe.sourceforge.net/index.html

[9] John Graham The spammers compendium.- 2003 Cambridge Spam Conference Proceeding - http://popfile.sourceforge.net

[10] Kristian Eide. Winning the War on spam: Comparison of Bayesian spam filters. 2003. http://home.dataparty.no/kristian/reviews/bayesian/

[11] Unam public spam set 2002-2003: http://www.seguridad.unam.mx/Servicios/spam/spam/

[12] From call for donations for this specific use, at the Universidad Carlos III de Madrid, 2003.

[13] Personal communication with Juan Carlos Martin, Security and Network Manager of EspacioIT, which has over 3.000 mail users along different domains and mail servers. October, 2003

[14] Carreras & Marquez. Boosting Trees for Antispam Email Filtering. 2001 TLAP Research Center. LSI Department. Universitat Politecnica de Catalunya.

[15] L.F. Cranor and B.A. LaMaochia. Spam Comunications of the ACM, 1998.

[16] Sholan M.Weiss and others. Maximizing text-mining performance. - 1999 IEEE Intelligents Systems.-

[17] Joachims. Text categorization with support vector machine. Proc. 10th Eur. Conf. Machine Learning. 1998.

[18] R.E. Schapire and Y.Singer. BoosText: boosting based system for categorization Machine Learning. 2000.

[19] Yang & Liu. A re-examination of text categorization methods. Proc. 22nd ADM SIGIR Conference 1999.

[20] W.Cohen Learning Rules for Classifying Mail. AAAI Spring Symposium on Machine Learning in Information Access. 1996.

[21] http://spamassassin.org/publiccorpus/ . - public - corpus AT jmason dot org if you have questions.

Health care and social inference systems: An unauthorized inference control based on fuzzy logic

Souhila Kaci[1], Abdeslam Ali-Laouar[2], and Frédéric Cuppens[2]

[1] Centre de Recherche en Informatique de Lens (C.R.I.L.–C.N.R.S.)
Rue de l'Université SP 16 62307 Lens France
kaci@cril.univ-artois.fr
[2] Institut de Recherche en Informatique de Toulouse (I.R.I.T-C.N.R.S)
118 route de Narbonne 62077 Toulouse France
{laouar,cuppens}@irit.fr

Abstract. In this paper, we address the problem of unauthorized inference of confidential information in the field of health care and social information systems. More precisely, we will focus on the problem of inference control of confidential information from statistical databases which contain information about patients and propopse a method based on fuzzy logic to avoid unauthorized inference. Information provided using our approach remains relevant because it is without loss of quality.

1 Introduction

The security of information systems is a very important problem which has been mainly addressed in military applications. This led to security policies which are applicable only in environments which accept a rigid bluk-heading of information and services handling this information. Indeed, these models cannot be used in other domains which also require security policies like for example the health care domain where it is important to guarantee the confidentiality, integrity and availability of pieces of information contained in medical files of patients. The confidentiality consists in expressing who has the right to reach which information about which, when, and possibly under which conditions. The integrity is the property which ensures that information is modified only by the users authorized under the conditions normally envisaged. Lastly, the availability is the aptitude of an information system for being able to be employed by the users competent under the conditions of accesses and use normally envisaged.

In this paper, we particularly address the problem of security of information systems in the field of health care and social. Let us note that in spite of the development of security policies in this context [6, 7], it is always possible for an external attacker and, especially, for an internal user badly disposed, to try to circumvent the mechanisms of access control to the resources in order to attack the confidentiality, the integrity or the availability of information.

To prevent the infringements against the intimacy of the patients, the medical databases must protect not only confidential information, but also information not explicitly confidential which can be employed to obtain confidential information. This paper treats detection and the limitation of the situations for which there is a risk of illegal inference (called also illegitimate inference). This problem is called *unauthorized inference problem*. It can also be simply defined in the following way. Suppose that a user is authorized to access to some information. The crucial question now is: can this user use this information to deduce a confidential information for which she would not have the right of access? A possible solution to this problem is to refuse to answer when this may allow to deduce confidential information however this solution is not interesting because it does not respect the availability condition. Another possible solution is the use of false answers for users having a restricted access to the information system. Indeed this method allows to protect confidential information by providing false but not very significant answers. The problem of this method is that the user to whom one provides false answers can make bad decisions. It is also difficult to provide a coherent set of false answers. The solution that we propose in this paper does not consist to provide a false answer to the user but a *"vague"* information formalized in *fuzzy logic* [8, 4].

Section 2 describes the problem of illegitimate information from databases containing information about the patients. We also describe a well-known method to attack such databases. In section 3, we first present the general principle of our approach. We then give some necessary background on fuzzy logic on which our approach is based. Lastly, section 4 gives a detailed description of our approach.

2 Illegitimate inference in statistical databases

The main difference between a statistical database (SDB for short) and a traditional one relates to the interrogation interface more limited in the SDB. The queries on a SDB are limited to operations like counting (COUNT), sum (SUM), the average (AVG) and other statistical calculus, which are carried out on subsets of data. Although these operations seem to be without consequence, it should be made sure that significant information on the individuals are not revealed. This problem becomes particularly difficult if we accept the possibility that a sequence of general queries, each one by itself does not allow to deduce confidential information, can be employed to deduce significant information. Let us now give an example to illustrate the difficult nature of the inference problem in the statistical databases. We consider a database, given in Table 1, which contains information concerning the employees. Let us suppose that the policy of the company imposes that the *salary* of the employees is a confidential information which should not be revealed. To achieve this goal, the database does not return an answer to a query like: *how much is the salary of the employee whose name is Isabelle?* since the answer

Table 1. Example of a statistical database.

Name	Sex	Age	Department	salary	Name	Sex	Age	Department	salary
Jean	M	27	Mathematics	2.000	Isabelle	F	27	Mathematics	2.600
Thomas	M	43	computer science	3.000	Justine	F	31	computer science	3.200

is confidential. Similarly, the base does not answer any query when, for example, the average is calculated on the basis of a simple record, i.e. a query concerning only one individual. Consequently, it refuses to answer for example the query: *how much is the average salary of the women employees who work for the computer science department?* because the average here is calculated from only one record.

A query on a SDB R consists to compute a subset of R using a characteristic formula C, which is a logical formula built from the values of the attributes of R by using the logical operators $\wedge$ (and), $\vee$ (or), and $\neg$ (not). For example, the subset of records representing the *women employees who work for the computer science department*, can be represented by the following characteristic formula:

$$C = (\textit{sex=F}) \wedge (\textit{department=computer science}).$$

The set of records which satisfy the characteristic formula C, denoted by X_C, is called the result of the query. Applying the formula C on the relation R given in Table 1, we get: $COUNT(C) = 1$, $AVG(Age, C) = 31$ and $SUM(Salary, C) = 3200$.

Generally, a statistical query taken separately does not allow to deduce confidential information. For this reason, a user with good intentions should be able to form any interesting characteristic formula, and to carry out any statistical measurement on the resulting set of the records. However, it is possible that a user forms statistical queries which can be employed to deduce specific values of a field of the database, which is not acceptable if the values represent confidential information. In this case, we say that the database has been compromised.

A characteristic formula used in order to compromise a database is called a tracker [2, 3]. This formula is chosen so that it gives as a result a set X_C whose size is equal to 1. Denning et col. [2] have shown that for any real database, a tracker can always be found.

In the next section, we propose a new strategy to prevent attacks based on trackers.

3 Our approach

In the everyday life and particularly in the medical field, medical analyses are generally expressed by linguistic descriptions (Example: Temperature of the body is raised, normal, etc). This is especially used for the non-specialists in the medical field. In this paper, we take as a starting point this method to deal with the illegitimate inference problem in statistical databases. More precisely, we replace the results of the statistical queries (quantitative answers) by linguistic descriptions (qualitative answers) in order to limit the risk of illegitimate inference.

For this, our idea consists in replacing the *numerical answers* (e.g. numbers of patients = 10) by *linguistic descriptions* (e.g. medium) formalized in fuzzy logic framework.

Intuitively, each numerical answer is associated to a given class then a qualitative answer is associated to each class. Thus, the formalization of our approach requires two steps: *classification* and *fuzzification*. Let us recall these two concepts:

- **Classification** is the procedure which consists in decomposing the scale of the used numerical values into non-empty classes so that each numerical value belongs to one and only one class.

  Let I be a set of elements. We say that $Q(I)$ is a partition of I if there exists a set

$\{q_1, q_2, \cdots, q_k\}$ satisfying the following conditions:

$\bigcup_{i=1,\cdots,k} q_i = I$ with $q_i \neq \emptyset$ and $q_i \cap q_j = \emptyset$ for $i \neq j$.

To be relevant, a partition should be made up of definitely individualized classes. Among existing classification methods, we recall one method, that we will use later, based on the aggregation around the centers using a fixed number of classes. The principle of this method is to determine a partition of I composed of k classes, the number k being fixed a priori by the user of the method. k centers $c_1, \cdots, c_k$ are chosen which are either arbitrarily points in the space of the variables, or elements of the set I.

Each element of the set I is associated to one and only one class whose center is one of the k centers $c_1, \cdots, c_k$ according to the following assignment rule:

i belongs to the class q_j of center c_j iff $||i - c_j|| = min_{l=1,\cdots,k}||i - c_l||$.

After the classification step, we have to associate an appropriate linguistic variable to each class. For example, if the numerical scale corresponds to the temperature then the linguistic variable which corresponds to the interval $[20, 25]$ may be *tepid*. This can be formalized in fuzzy logic [8].

- **Fuzzification:** A principal characteristic of the human reasoning is that it is based on vague or incomplete data. Thus, to determine if a temperature is hot or cold is easy for any individual without necessarily knowing its exact value. Fuzzy logic has the aim of studying the representation of vague knowledge and the approximate reasoning. A principal characteristic of fuzzy logic is that an object may belong to a set and at the same time to its complement. Thus, a temperature of 22 may at the same time be hot and not hot.

  A *linguistic variable* is a triple (X, V, F_X), where X is a variable (age, temperature, etc) defined on a set of reference V (the set of integers, reals, etc). $F_X = \{A_1, A_2, \cdots\}$ is a finite or infinite set of subsets of V used to characterize X (old, young, hot, cold, etc). Each fuzzy subset represents a linguistic description.

  The variable may belong to one or more subsets of this element of reference. For example, the temperature $T = 28$ may belong to the subset "pleasant" but may also belong partly to the subset "hot".

  The membership relation between a variable and a subset is called *membership function*. In other terms, we speak about the membership degree of a variable x to a subset F, denoted by $\mu_F(x)$.

  A *fuzzy set F* of universe Ω (a fuzzy subset of Ω) is defined by a membership function μ_f which associates to each element x of Ω a value in the interval $[0, 1]$.

  $$\begin{aligned} \mu_F : \Omega &\rightarrow [0, 1] \\ x &\mapsto \mu_F(x) \end{aligned}$$

  $\mu_F(x)$ represents the membership degree of x to the set F. By definition, if $\mu_F(x) = 0$ then x does not belong to F and more $\mu_F(x)$ approaches 1, more the value x belongs to F. If $\mu_F(x) = 1$ then x belongs completely to F.

  A fuzzy subset is said to be convex if and only if:

  $$\forall x, y; x > y, \forall z \in [x, y], \mu_F(z) \geq min(\mu_F(x), \mu_F(y)).$$

  Generally, we express numerical quantities by vague linguistic descriptions such as "approximately 100". The results of fuzzy measurements or an error analysis are modelled by fuzzy sets called *fuzzy quantities*. A fuzzy quantity Q is a fuzzy set in the universe $\mathbb{R}$ of real numbers. It is supposed to be normalized.

A *fuzzy interval* N is a convex fuzzy quantity. It is a generalization of a real interval whose extremities are fuzzy in order to model concepts such as ”approximately”, ”roughly”, etc.

– Representation of a L-R fuzzy interval A fuzzy interval of type LR has a membership function built from a quadruplet $A = (m_1, m_2, a, b)$, where m_1, m_2, a and b are strictly positive real numbers, and of two functions L and R from $\mathbb{R}^+$ into the interval $[0, 1]$ semi-continuous, non-increasing and satisfying the conditions:

- $L(0) = R(0) = 1$,
- $L(1) = 0$ or $\forall x \in \mathbb{R}^+, L(x) > 0$ and $lim_{x\to+\infty}L(x) = 0$,
- $R(1) = 0$ or $\forall x \in \mathbb{R}^+, R(x) > 0$ and $lim_{x\to+\infty}R(x) = 0$.

The membership function is defined as follows:

$$\mu_F(x) = \begin{cases} L(\frac{m_1-x}{a}) \text{ if } m_1 - a \leq x \leq m_1 \\ \quad 1 \text{ if } m_1 < x < m_2 \\ R(\frac{x-m_2}{b}) \text{ if } m_2 \leq x \leq m_2 + b \\ \quad 0 \text{ if } x < m_1 - a \text{ or } x > m_2 + b \end{cases}$$

When $m_1 = m_2 = m$, the fuzzy interval $P = (m, m, a, b)_{LR}$ is called *a fuzzy number*, denoted by $P = (m, a, b)_{LR}$ and whose membership function is defined as follows:

$\mu_F(x) = L(\frac{m-x}{a})$if $x < m$, $\mu_F(x) = 1$if $x = m$ and $\mu_F(x) = R(\frac{x-m}{b})$if $x > m$.

Let $P_1 = (p_1, \alpha_1, \beta_1)_{LR}$ and $P_2 = (p_2, \alpha_2, \beta_2)_{LR}$ be two LR-fuzzy numbers. Then the addition $\oplus$, the substraction $\ominus$ and multiplication $\otimes$ are defined by [4]:

$P_1 \oplus P_2 = (p_1 + p_2, \alpha_2 + \alpha_2, \beta_1 + \beta_2)_{LR}$.

$P_1 \ominus P_2 = (p_1 - p_2, \alpha_1 + \alpha_2, \beta_1 + \beta_2)_{LR}$.

$P_1 \otimes P_2 = (p_1 * p_2, p_1 * \alpha_2 + p_2 * \alpha_1, p_1 * \beta_2 + p_2 * \beta_1)_{LR}$.

Contrary to the addition and subtraction, the multiplication $P_1 \otimes P_2$ is not of type LR. An approximate value of type LR is given when P_1 and P_2 have a support included in $\mathbb{R}^+$, α_1 and β_1 are small w.r.t. p_1 and, α_2 and β_2 are small w.r.t. of p_2.

To apply a linguistic representation to a quantitative variable, the principle consists in breaking up all possible values of the given quantitative variable into subsets (a set of classes of values), so that the borders of the classes are not clearly given. This treatment allows to transform a numerical input into a fuzzy subset. The decomposition should not be arbitrary but founded on criteria, such as the homogeneity of the classes, the uniform partition of the universe, the subsets are totally ordered. These subsets are also called *”linguistic variables”*.

The subsets are characterized by their associated membership functions; we associate a membership function to each subset. Their positions and overlappings can be chosen arbitrarily provided that the following conditions are verified: their form should be convex, the subsets (often in the form of trapezoid) should be partially overlapped so that there are no unspecified ranges and lastly to avoid to imbricating more than two subsets.

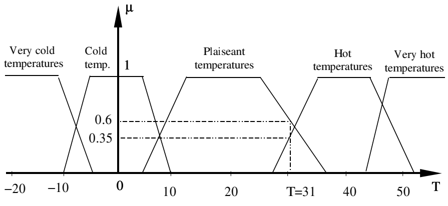


Fig. 1. Representation of the temperature in fuzzy logic.

Example 1. Let us consider the temperature input $T = 31$. According to the membership function given in Figure 1, we obtain the following values:
μ_T(very cold temperatures) $= \mu_T$(cold temperatutres) $= 0$, μ_T(pleasant temperatures) $=$ .6, μ_T(hot temperatures) $= .35$ and μ_T(very hot temperatures) $= 0$.

Now, it seems important to answer some questions : How many *classes* is it necessary to represent each quantitative variable? Which are the best *linguistic values* for each class? For the first question, more the number of linguistic values is high, more the partitioning quality is good. It is necessary however that the rate: $Card(\Omega)/$*Number of Partitions* is not equal to 1, otherwise this simply means that there is no fuzzification.
For the second question, we compute the membership degree of each element x to all the subsets F_i of the universe Ω. Let $\mu_{F_i}(x)$ be the membership degree of x to F_i. We say that $x \in F_i$ only if $\forall F \in \Omega, \mu_F(x) \leq \mu_{F_i}(x)$.

4 Detailed description of our approach

The principle of our method consists, in a first step, to decompose the set of values of the confidential attributes into subclasses of values. Each subclass contains values according to a given criterion. In this paper, we will use the classification method based on a fixed number of classes.
After the classification into subclasses the fuzzification comes. We transform each class into a fuzzy quantity i.e., a fuzzy number with a membership function. Then, we associate a linguistic variable to each number (small, large, etc). Next, for each answer provided by the database management system, we compute the membership degree of this answer to each fuzzy subset (linguistic variables). The answer of our system is the linguistic variable which has the highest membership degree.
Let us note that the simplest version of a statistical query SQL is written as follows:

SELECT f(<attributes>) *FROM* <relations> *WHERE* <conditions>,

where f is a statistical function such as Sum, Avg, $Count$, etc.

In this paper, we focus on queries which compute *statistical quantities*, i.e. queries which deduce information on aggregation such as sum, $average$, max and min.
Let us consider the example of relation R (patient, H/F, age, sickness insurance company, leucocyte rate) given in the Table 2 (borrowed from [5]).
The number of patients is 10 and the normal leucocyte rate in mm3 of blood is 4500. In this example, we suppose that the *leucocyte rate* is a confidential attribute. To control the illegitimate inference on this attribute, we will transform the answers to the queries

Table 2. Example of a database.

Patient	M/F	Age	Sick. ins.	Leucocyte	Patient	M/F	Age	Sick. ins.	Leucocyte
Dufour	M	45	MAAF	4000	Dupont	M	30	MMA	6000
Dulac	F	35	MMA	7000	Dupr	F	32	IPECA	7200
Dulon	M	55	MGEN	3500	Dupuis	F	50	MGEN	6800
Dumas	M	40	Rempart	3800	Durand	F	25	LMDE	3000
Dumont	M	38	MMA	7500	Duval	M	45	IPECA	5500

concerning this attribute by giving qualitative answers.
We proceed in the same way for the answers to the queries which compute the number of patients who verify a given condition. For this, we fuzzify the number of patients and the leucocyte rate.
Let us start with the number of patients and decompose this variable as follows: A first class: from 0 to 3, a second class: from 4 to 6 and a third class: from 7 to 10.
We now transform each class into a fuzzy number $A_i(m, a, b)$ where m is the center of the class, a and b represent the degrees of inaccuracy.
For each number, we associate a linguistic variable (see also Figure 2-a):

- The first class is fuzzified by the fuzzy number ”$small$” $= (2, 2, 2)_{LR}$,
- The second class is fuzzified by the fuzzy number ”$medium$” $= (5, 2, 2)_{LR}$,
- The third class is fuzzified by the fuzzy number ”$great$” $= (8, 2, 2)_{LR}$.

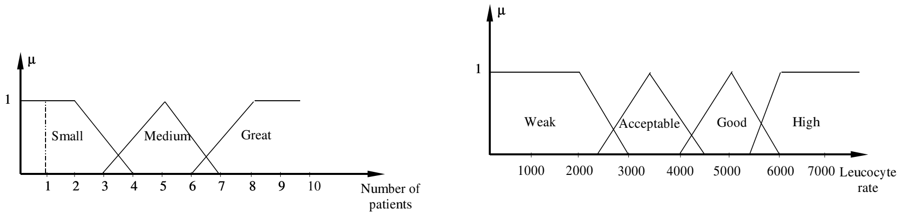


Fig. 2. (a) Fuzzification of the number of patients. **(b)** Fuzzification of the leucocyte rate.

We now classify the leucocyte rate for a patient as follows:

- 1^{st} class: from 0 to 3000, 2^{nd} class: from 3000 to 4500,
- 3^{rd} class: from 4500 to 6000 and 4^{th} class: from 6000 to 7000.

We now propose the following fuzzification (see also Figure 2-b):

- The first class is fuzzified by the fuzzy number ”$weak$” $= (2000, 1000, 1000)_{LR}$
- The second class is fuzzified by the fuzzy number ”$acceptable$” $= (3500, 1000, 1000)_{LR}$
- The third class is fuzzified by the fuzzy number ”$good$” $= (5000, 1000, 1000)_{LR}$
- The fourth class is fuzzified by the fuzzy number ”$high$” $= (6000, 1000, 1000)_{LR}$

Let us now suppose that a user is authorized to carry out statistical queries and she wants to discover the leucocyte rate of ”Dulon”. Let us also suppose that this user knows moreover that ”Dulon” has the MGEN as a sickness insurance company. Consider now the following queries:

1) SELECT Count(Patient) FROM R WHERE M/F=’M’ AND Sick. ins. =’MGEN’

$Result = 1$ (R_1)

Let us compute the membership degrees $\mu_{F_i}(R_1)$ of the result (R_1) w.r.t. each fuzzy subset. We get: $\mu_{small}(R_1) = 1$, $\mu_{medium}(R_1) = 0$ and $\mu_{great}(R_1) = 0$. So the answer provided after fuzzification is "small" since it corresponds to the highest membership degree.

2) SELECT AVG(Leucocyte) FROM R WHERE M/F='M' AND Sick. ins. = 'MGEN'

$Result = 3500$ (R_2)

We compute the membership degrees $\mu_{F_i}(R_2)$ of the result (R_2) w.r.t. each fuzzy subset: $\mu_{weak}(R_2) = \mu_{good}(R_2) = \mu_{high}(R_2) = 0$ and $\mu_{acceptable}(R_2) = 1$. Then the answer is "acceptable".

Note that the deduction of confidential information when handling numerical answers is very easy. It is clear that from (R_1) and (R_2), the user may directly deduce that the leucocyte rate of "Dulon" is equal to 3500.

The case of the qualitative answers is less simple: from (R_1), the user knows that the size of the set to which "Dulon" belongs is "small", and from (R_2), she deduces that their average of the leucocyte rate (the set "small") is "acceptable".

Let us now see what may the user deduce from these two information. For this, we know that the average is defined by the equation $\overline{x} = \frac{1}{n} \sum x_i$. It is clear that when n is equal to 1, the average is equal to x_i. To see the impact of the fuzzification on the reasoning of the user, we will analyze the use of the fuzzification step by step:

- Let us suppose that the number of patients is not fuzzified whereas the leucocyte rate is. The answer given to the user in this case is then: the number of patients is equal to 1 (as an answer to the query (R_1)) and their average leucocyte rate is "acceptable", which allows to deduce that the leucocyte rate of Dulon is "acceptable". However, the fuzzification of the number of patients makes that the answer provided to the user (also as an answer to the query (R_1)) is "small", which does not allow to know precisely how many patients correspond to this answer "small".
- Let us now suppose that the user knows moreover that the maximum size of the fuzzy quantity "small" is equal for example to two. However even if the user has this information, she deduces nothing as we will show on the following example: It is known that an "acceptable" leucocyte rate lies between 3000 and 4500. Let the size of "small" be equal to 2. From a leuycocyte average of two patients x_1 and x_2 equal to "acceptable", we may have the following possibilities for x_1 and x_2:
  - $x_1 = 2500 \equiv$ "weak"[3], $x_2 = 4000 \equiv$ "acceptable"
  - $x_1 = 2500 \equiv$ "weak", $x_2 = 5000 \equiv$ "good"
  - $x_1 = 1500 \equiv$ "weak", $x_2 = 6500 \equiv$ "high"
  - $x_1 = 3500 \equiv$ "acceptable", $x_2 = 5000 \equiv$ "good"
  - $x_1 = 3500 \equiv$ "acceptable", $x_2 = 4000 \equiv$ "acceptable".

  From these results, one can say that from a leucocyte average equal to "acceptable" computed for two patients, one concludes nothing on the leucocyte rate of one of the two patients.

[3] The equivalence means here that the number (e.g. 2500) corresponds to the given class (e.g. "weak") after fuzzification.

Note that to have an average rate "acceptable", we have five possibilities for the leucocyte rate for each of the two patients. In only one case, the rate of the two patients is "acceptable". In the other cases, it varies between "Weak", "acceptable", "Good" and "High". So we have four cases with x_1 or x_2 equal to "acceptable" and six cases different from "acceptable".

Then we may say that it is totally possible that the leucocyte rate of "Dulon" is equal to "acceptable", but it is also totally possible that it is different from "acceptable". Indeed, we are in a situation of total ignorance.

Let us note that in the real case, the database may contain thousands of patients and the fuzzy quantity "small" may reach several hundreds of patients. Consequently, the possibilities of inference are even weaker when the cardinality which corresponds to the fuzzy quantity is larger. The user deduces nothing on the leucocyte rate of "Dulon" when all the possible cases are considered.

3) SELECT Count(Patient) FROM R. Then, $Result = 10$ (R_3)

The answer after fuzzification is "great" (R'_3)

The user tries thereafter to know the number of patients different from "Dulon". For this, she gives the following query:

4) SELECT Count(Patient) FROM R WHERE NOT (M/F='M' AND Sick. ins.='MGEN'); $Result = 9$ (R_4)

The answer after fuzzification is "great" (R'_4)

From these two answers, the user may construct the following reasoning: The difference between (R_3) and (R_4) (10-9=1) corresponds to the number of male patients who have the MGEN as a sickness insurance company (i.e., the number of patients having the same properties as "Dulon").

With a similar reasoning, she concludes that the difference between (R'_3) and (R'_4) is equal to [4] $\mid "great" \ominus "great" \mid = \mid (8,2,2)_{LR} \ominus (8,2,2)_{LR} \mid = \mid (8,2,2)_{LR} \oplus (-8,2,2)_{LR} \mid = \mid (0,4,4)_{LR} \mid$ which is equivalent to $(0,0,4)_{LR}$ after removing the negative part, since there is no negative leucocyte rate.

So we have $(R'_3) \ominus (R'_4) \sim "small"$. Indeed, we get the same result as for (R_1) after fuzzification.

To know the average of the leucocyte rate for all the patients, the user gives the following query:

5) SELECT AVG(Leucocyte) FROM R. Then, $Result = 5430$ (R_5)

The answer after fuzzification is "good" (R'_5)

To compute the average of the leucocyte rate of all the patients different from "Dulon", the user gives the following query:

6) SELECT AVG(Leucocyte) FROM R WHERE NOT (M/F='M' AND Sick. ins.='MGEN'), $Result = 5644$ (R_6)

The answer after fuzzification is "high" (R'_6)

In the case of numerical answers, to know the leucocyte rate of "Dulon", the user computes the following value: $10 * 5430 - 9 * 5644 = 3500$.

With a similar reasoning, in the case of qualitative answers, she may try to proceed

[4] Since the values are not known a priori but supposed to be positive, the subtraction is translated into fuzzy logic by the absolute value.

in the following way. The leucocyte rate of ”Dulon” is equal to:
$\mid ((R'_3) \otimes (R'_5)) \ominus ((R'_4) \otimes (R'_6)) \mid \sim\mid (-8000, 38000, 38000)_{LR} \mid$.
From the obtained number, the user deduces nothing because the leucocyte rate is never negative. Even if she can deduce some information (if the fuzzification is changed), the situation is similar to the first case since the user does not know the exact number of patients. Let us also note that we lost the precision on the computation of the leucocyte rate because of the multiplication which we carried out (recall that in the case of the multiplication, the computation is only approximate).

We have shown on this example that the user may use different ways to deduce confidential information however the use of qualitative answers makes difficult the implementation of attacks by trackers because after fuzzification, it is difficult to identify the individual concerned by the confidential information. Indeed, required information is not distinguished after fuzzification.

5 Conclusion

We have proposed a first attempt to limit the risk of inference of confidential information from a database using fuzzy logic. It is difficult to affirm here that we eliminate any risk of illegal inference. The goal is nevertheless to continue to answer the queries as well as possible using non-confidential information. So our aim is to limit at least as possible the restrictions of legitimate access on databases while ensuring that the risk of unauthorized inference remains below an acceptable threshold.

An immediate prospect for this work would be to implement our approach and to validate it on great databases. We showed in this paper that our approach particularly enables us to control the attacks by trackers. We expect to see how this approach could be used to control other types of attacks like linear systems [2, 1]. Lastly, it would be interesting to see to what extent our approach is sensitive to the classification method used, i.e. to see if the use of other classification methods give sensitively different results.

References

1. F. Cuppens. A logical analysis of authorized and prohibited information flows. In IEEE Symposium on Research in Security and Privacy, 1993.
2. D. Denning, P. Denning and Schwartz. The tracker: A Threat to Statistical Database Security. ACM Transactions on Database Systems, 4(1): 76-96, 1979.
3. D. Denning and J. Schlorer. A Fast Algorithm for Calculating a Tracker in Statistical Databse. ACM Transactions on Database Systems, 5(1), 1980.
4. D. Dubois and H. Prade. La logique floue. In Quaderni, 50-73, 1995.
5. A.A. El Kalam. MP6, Sous-projet 3: Politiques de scurit pour les SICSS. Informations protger et menaces. Rapport technique.
6. R. Sandhu, E. Coyne, H. Feinstein and C. Youman. Role-based access control models. IEEE Computer, 29:38-47, 1996.
7. S. Solms. The management of computer security profiles using a role-oriented approach. Computer and Security, 13(8), 673-680, 1994.
8. L. Zadeh. Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems, 1, 3-28, 1978.

An XML framework for multi-level access control in the enterprise domain

Ioannis Priggouris[1], Stathes Hadjiefthymiades[1], Lazaros Merakos[1]

[1] University of Athens, Department of Informatics & Telecommunications,
Panepistimioupolis, Ilissia, 15784,
Athens, Greece
{iprigg, shadj, merakos}@di.uoa.gr

Abstract. Modeling security information has always been a fundamental part of every security system. A robust and flexible model is needed in order to guarantee both the easy management of security information and the efficient implementation of security mechanisms. In this paper, we present an XML-based framework, which can be used for controlling access to computer systems. The framework is mainly targeted to enterprise systems and aims to provide a fine-grained access control infrastructure for securing access to hosted services. The proposed framework supports both role-based and user-based access control on different levels. Although, the discussion focuses mainly on the data model, access control schemes and guidelines for implementing fitting security architectures are also provided.

1 Introduction

Secure service access comprises an area of extensive research and interest in the recent years. Different mechanisms and techniques have been adopted with the purpose of securing access to computer systems. However, apart from the implementation of the security mechanism a crucial issue in designing a robust security framework is the structure of the security meta-information, which is consulted in order to verify eligibility of a user for entering the system.

In this paper we present a framework, which can be used for implementing authentication and access control mechanisms over heterogeneous IT infrastructures. The framework defines the data structures needed for storing security information, as well as the actual process for implementing authorization and controlling access to specific resources. The data model specified using XML [7], which makes the architecture portable over different information repositories (xml files, RDBMSs, Directory Services, etc.). Our architecture is targeted to enterprise systems hosting multiple services. Its design is focused on providing a flexible scheme, which could sufficiently support such multi-service environments.

The rest of the paper is structured as follows. In section 2 a brief overview of related work and limitations of existing access control schemes is presented, followed by the description of our proposed model in section 3. The architectural aspects of the security infrastructure follows in section 4 and the paper concludes with a summary of the innovation achieved and a discussion on its potential application domain.

2 Technology overview and related work

The simplest form of access control is the client authentication mechanism, which, however in its primitive form provides a flat security model. Nevertheless it can be augmented, with support for roles, in order to provide a multi-level security model, where access to individual resources is controlled separately. A role is an internal identity of the system, which defines the resources that a specific user is allowed to access. Role-based security is an elegant way to provide user authorisation and user access checks for an application. A user belonging to a particular role can access code, software and resources for which permissions are granted to the role. Incorporating roles makes security management much more flexible, while the security framework is rendered capable of supporting different security levels.

Role Based mechanisms for securing access to resources attracted significant research interest after 1990, when the concept of Role-Based Access Control (RBAC) emerged rapidly as a proven technology for managing and enforcing security in large-scale systems. A significant number of research papers on RBAC models and experimental implementations has been published in the recent past [1], [2], [3], [4], [5], [6]. A certain shortcoming of all these models is that they define RBAC mechanisms based on the assumption that roles have global scope. This assumption makes them inadequate for large enterprise environments, hosting multiple services, which are administered from different vendors. In such environments, using global roles is not advisable as their management may prove significant problem for the potential administrator, especially if the number of hosted services increases substantially. Evidently, a more flexible approach for controlling access to the resources hosted by such systems is needed.

3 Security Model

The model we propose is much more fine-grained than those available today. Each service defines specific roles, which are authoritative only within its context, having no impact on other services. Moreover, as discussed below, our architecture achieves all the above without restricting the potential namespace of the roles or the services. These characteristics are ideal for service provisioning platforms or other systems hosting varied functional entities, as it reduces drastically the administrative overhead needed for managing security roles. Moreover the model allows distributed management schemes to be adopted both for roles definition and for security enforcement. In such schemes, separate administrative entities can be responsible for specifying roles within a single service and assigning users to them, without caring if these roles have already been specified inside other services also.

Before delving further into the design aspects of the framework we will try to formalise it using propositional calculus. Our aim is to provide the basis for the design work that follows as well as a notation reference for future research in the same area. Similar formal approaches have been introduced in the past for equivalent architectures, such as the OASIS role-based access control framework [12]. Definitions of basic concepts, like *services*, *methods*, *roles* and *users,* which will help the reader understand better the security architecture are also presented.

The model is based upon 6 basic sets:

- *U*: set of users
- *S*: set of services
- *M*: set of methods
- *G*: set of method signatures
- *R*: set of roles
- *N*: set of role names

A simple user *u* is an element of U ($u \in U$) and is defined as an entity, interacting with the protected computer system. The user usually is a human; however client programs or other computer systems can also be considered as users.

A service *s* is an element of S ($s \in S$), and corresponds to a software component running on a computer system. Borrowing the Object oriented terminology the service is the equivalent of an object and consists of several methods, which are the actual resources that need to be protected; since no other access in allowed to the service entity. Each method has a signature $g \in G$, which consists of its name and the list of invocation parameters. We won't delve further into the definition of the method signature, as it is not of prime importance to our model. A significant constraint of the model, we have presented so far, is that method signatures, although unique within the scope of each service, are not unique within the computer system. In order to surpass this constraint we use the pair $(s,g) \in S \times G$ introducing the concept of method *m* as an element of M ($m \in M$), which apparently bears global uniqueness because $M \subseteq S \times G$. Moreover, we denote as M_s the subset of methods belonging to the same service *s*. Evidently $M_s = \{m_1, m_2, ...m_k\}$ where $m_i \in M$ for $1 \le i \le k$ and $M_s \subseteq \{s\} \times G$ for each $s \in S$.

A role name *n* is an element of N ($n \in N$) and defines a logical label, which is used within a computer system for diversifying access to the hosted services. A role name is unique within the scope of its defining service. However, it can be re-used in the context of another service. In order to avoid confusion between the two definitions our model uses the pair $(s,n) \in S \times N$ in order to define a globally unique role $r \in R$.

In order to achieve the objective of protecting critical resources each role *r* is associated both with methods and users. Association with methods is used in order to determine the resources to be protected and will be hereafter referred to as *role declaration*. Association with users, on the other hand, defines the access rights to the method and will be referred as *role assignment*. In a more formal notation, a *role declaration* corresponds to a pair of $(r,m) \in R \times M$, while a *role assignment* to another pair $(r,u) \in R \times U$. In order for the user *u* to have access to a certain service method *m* both a *role declaration* and a *role assignment* for the same role *r* must have been defined within the model. Another association that can be defined in our model is that between users and services. This association, which will be referred to as *service eligibility,* is expressed in the form of pairs of $(s,u) \in S \times U$ and indicate that a user *u* is eligible to access the service *s*.

4 Architecture- Framework design

In this section we provide the foundations of our architectural approach and issues considered during the design phase. As already mentioned the security framework, supports the following 3 basic security operations:

- Authentication,
- Role-Based Access Control (RBAC)
- User-Based Access Control (UBAC)

Authentication is not actually covered by our model, but it is used in order to determine the identity of a user. The authentication mechanism undertakes the task of establishing a security context, which will carry all the privileges assigned to the specific user (e.g., roles). Of course these roles are specified inside the framework and should somehow be mapped to the specific application domain (e.g., the services). However this is an implementation specific issue, which should be considered when implementing the discussed framework. The simplest way to achieve this mapping is by hard-coding them inside the applications. Enterprise software offers alternative much more flexible ways, by using deployment descriptors ([9], [10]).

User-based access control is supported in two different levels:

- A low-level access control, which enables controlled access to the whole infrastructure.
- A high-level access control, providing a more fine-grained mechanism, which allows controlling access to a specific set of resources (i.e., a single service).

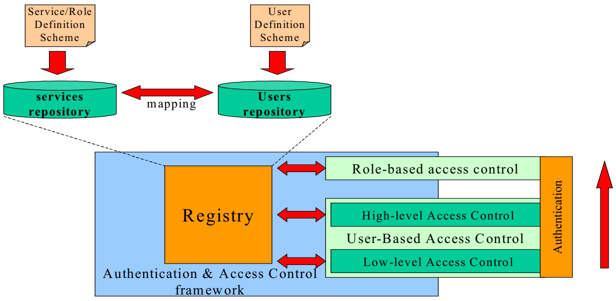

Service/Role Definition Scheme
User Definition Scheme
services repository
mapping
Users repository
Registry
Role-based access control
High-level Access Control
User-Based Access Control
Low-level Access Control
Authentication
Authentication & Access Control framework

Fig. 1. Overall architecture

Role-based and user-based access control work independently of each other but they both rely on successful user authentication. Depending on the pursued functionality, the framework can be configured to enforce role-based or user-based access control only. The mechanisms could also stack in order to provide an integrated multi-level access control infrastructure. The proposed stack order is defined by the arrow in figure 1; low-level user-based access control is the most coarse security mechanism so it is the first to be invoked while role-based the most refined one and therefore it is placed last.

Delving inside the heart of the security architecture we find the Registry module. The Registry holds all information pertaining to potential users of the system, running services and their roles. Moreover, the mapping between users and roles is harbored in it. The Registry is updated every time a new service (i.e., bundle of resources) is installed on the protected system. It is also updated each time a new user is defined as well as whenever the association between users and services is redefined. In subsequent sections the internal structure of the Registry will be presented in detail along with the key aspects that differentiate its design and allow it to fit in dynamic multi-service environments.

4.1 Registry

The Registry accommodates two repositories: one for services and roles and another for the users. Each Repository contains a set of entries of the same type. In order to populate the two repositories, specific schemes defining the structure of each stored element were designed. Specifically we defined:

- The User definition scheme
- The Service/Role definition scheme

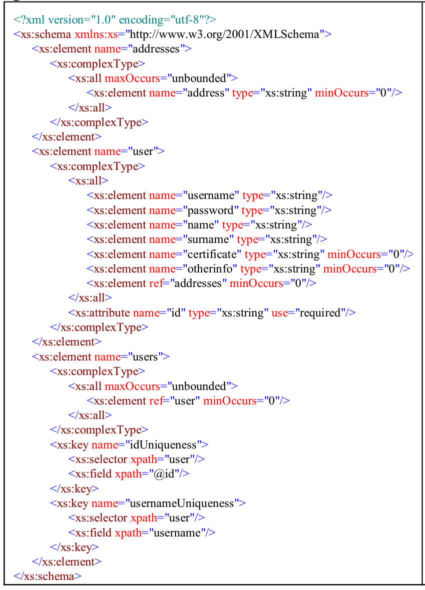
Other information contained in the Registry includes the mapping between services, roles and users.

User definition scheme. This scheme specifies the way a user entry is stored. User entries act as a container for user-specific data. The defined scheme is fairly simple and can be seen in figure 2. Although the specification comes in the form of an XML schema [8], the presented framework does not consider any particular implementation. Thus, possible implementations may include xml files, RDBMS tables and LDAP objects.

Each user entry is identified by the unique *id* attribute and also has a unique *username* value. The framework uses the *id* attribute for internally discriminating between users, while the *username* is an easily memorized alias of the *id.* The scheme also defines an optional element for storing certificates, which can be used for supporting certificate-based client authentication. The rest of the fields (*name*, *surname*, *other-info, addresses* etc.) are rather trivial and are mainly used for storing supplementary information for each user.

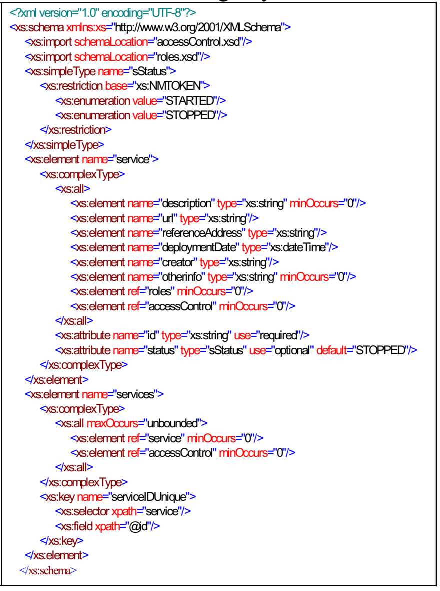
Service/Role definition scheme. The Service definition scheme specifies the structure of the service entry, which provides a convenient storage scheme for service-specific data. The scheme can store various information elements pertaining to the service, as seen in figure 3. The existing information elements were adopted in order to apply the security framework in a service provisioning platform, where services were exposed through a web interface. However, the exact definition of the service scheme can vary according to the application domain as other applications may require additional data to be stored or render some of the existing elements obsolete.

The specification of the *roles* element is presented in figure 4. Individual roles are identified by an *id* attribute. The *id* corresponds to the role name (*n*), as defined in the formal model, whose uniqueness within the scope of the same service is enforced by the proposed service specification. Embedding each role inside the service entry allows for the automatic pairing between service and role *ids* (i.e, the (*s,n*) pairs identifying the globally unique role *r*). A status attribute is supported for each role, allowing its enablement or disablement on-demand. The *role* is also the entity, which contains the actual association with the users (i.e. the *role assignment* that was defined in the formal model). In order to avoid duplicate *member* entries for each role, the corresponding element is marked as unique. The values of the *member* elements correspond to the *ids* of the users as the latter are defined inside the registry.



```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="addresses">
        <xs:complexType>
            <xs:all maxOccurs="unbounded">
                <xs:element name="address" type="xs:string" minOccurs="0"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:element name="user">
        <xs:complexType>
            <xs:all>
                <xs:element name="username" type="xs:string"/>
                <xs:element name="password" type="xs:string"/>
                <xs:element name="name" type="xs:string"/>
                <xs:element name="surname" type="xs:string"/>
                <xs:element name="certificate" type="xs:string" minOccurs="0"/>
                <xs:element name="otherinfo" type="xs:string" minOccurs="0"/>
                <xs:element ref="addresses" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="id" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="users">
        <xs:complexType>
            <xs:all maxOccurs="unbounded">
                <xs:element ref="user" minOccurs="0"/>
            </xs:all>
        </xs:complexType>
        <xs:key name="idUniqueness">
            <xs:selector xpath="user"/>
            <xs:field xpath="@id"/>
        </xs:key>
        <xs:key name="usernameUniqueness">
            <xs:selector xpath="user"/>
            <xs:field xpath="username"/>
        </xs:key>
    </xs:element>
</xs:schema>
```

Fig. 2. User entry specification

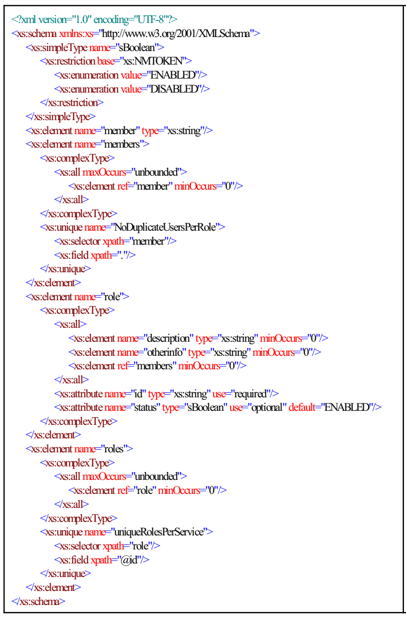


```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:import schemaLocation="accessControl.xsd"/>
    <xs:import schemaLocation="roles.xsd"/>
    <xs:simpleType name="sStatus">
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="STARTED"/>
            <xs:enumeration value="STOPPED"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="service">
        <xs:complexType>
            <xs:all>
                <xs:element name="description" type="xs:string" minOccurs="0"/>
                <xs:element name="url" type="xs:string"/>
                <xs:element name="referenceAddress" type="xs:string"/>
                <xs:element name="deploymentDate" type="xs:dateTime"/>
                <xs:element name="creator" type="xs:string"/>
                <xs:element name="otherinfo" type="xs:string" minOccurs="0"/>
                <xs:element ref="roles" minOccurs="0"/>
                <xs:element ref="accessControl" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="id" type="xs:string" use="required"/>
            <xs:attribute name="status" type="sStatus" use="optional" default="STOPPED"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="services">
        <xs:complexType>
            <xs:all maxOccurs="unbounded">
                <xs:element ref="service" minOccurs="0"/>
                <xs:element ref="accessControl" minOccurs="0"/>
            </xs:all>
        </xs:complexType>
        <xs:key name="serviceIDUnique">
            <xs:selector xpath="service"/>
            <xs:field xpath="@id"/>
        </xs:key>
    </xs:element>
   </xs:schema>
```

Fig. 3. Service entry specification

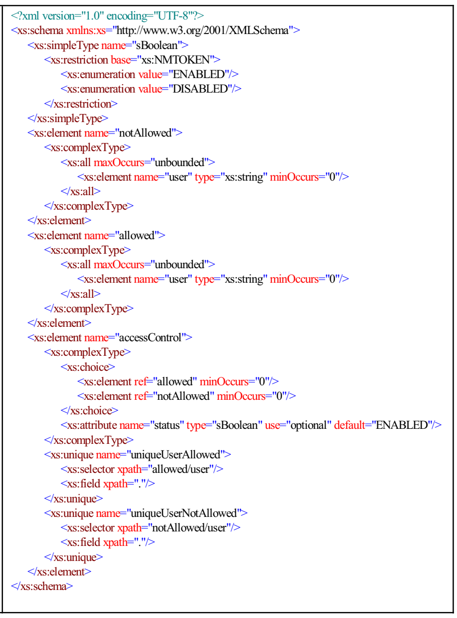
The service specification contains also the appropriate information needed by the framework's access control mechanisms. Linking to these mechanisms is achieved through the defined *accessControl* element. The aforementioned element appears in two different levels within the service specification (see figure 3); one at the *services* level, which intends to cover low-level access to all possible resources (i.e., to the whole protected system) and a second one at the *service* level. The latter realises the *service eligibility* association, defined in our model, thus implementing the high-level access control that was defined in the beginning of the section.

The defined schema for the *accessControl* element is presented in figure 5. In order to support a flexible definition framework, the schema has the option of choosing between specifying either a list of users eligible to access the controlled resource or a list of non-eligible users. The appropriate information is stored under the *allowed* or *notAllowed* elements respectively, in the form of sets of user *ids*. The proposed scheme validates that the same *user* does not appear sin two different areas inside the same *accessControl* element, thus avoiding erroneous situations, where two conflicting restrictions apply to a single user.



```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:simpleType name="sBoolean">
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="ENABLED"/>
            <xs:enumeration value="DISABLED"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="member" type="xs:string"/>
    <xs:element name="members">
        <xs:complexType>
            <xs:all maxOccurs="unbounded">
                <xs:element ref="member" minOccurs="0"/>
            </xs:all>
        </xs:complexType>
        <xs:unique name="NoDuplicateUsersPerRole">
            <xs:selector xpath="member"/>
            <xs:field xpath="."/>
        </xs:unique>
    </xs:element>
    <xs:element name="role">
        <xs:complexType>
            <xs:all>
                <xs:element name="description" type="xs:string" minOccurs="0"/>
                <xs:element name="otherinfo" type="xs:string" minOccurs="0"/>
                <xs:element ref="members" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="id" type="xs:string" use="required"/>
            <xs:attribute name="status" type="sBoolean" use="optional" default="ENABLED"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="roles">
        <xs:complexType>
            <xs:all maxOccurs="unbounded">
                <xs:element ref="role" minOccurs="0"/>
            </xs:all>
        </xs:complexType>
        <xs:unique name="uniqueRolesPerService">
            <xs:selector xpath="role"/>
            <xs:field xpath="@id"/>
        </xs:unique>
    </xs:element>
</xs:schema>
```

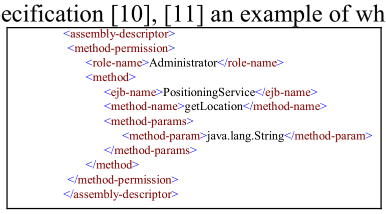
Fig. 4. Roles specification



```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:simpleType name="sBoolean">
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="ENABLED"/>
            <xs:enumeration value="DISABLED"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="notAllowed">
        <xs:complexType>
            <xs:all maxOccurs="unbounded">
                <xs:element name="user" type="xs:string" minOccurs="0"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:element name="allowed">
        <xs:complexType>
            <xs:all maxOccurs="unbounded">
                <xs:element name="user" type="xs:string" minOccurs="0"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:element name="accessControl">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="allowed" minOccurs="0"/>
                <xs:element ref="notAllowed" minOccurs="0"/>
            </xs:choice>
            <xs:attribute name="status" type="sBoolean" use="optional" default="ENABLED"/>
        </xs:complexType>
        <xs:unique name="uniqueUserAllowed">
            <xs:selector xpath="allowed/user"/>
            <xs:field xpath="."/>
        </xs:unique>
        <xs:unique name="uniqueUserNotAllowed">
            <xs:selector xpath="notAllowed/user"/>
            <xs:field xpath="."/>
        </xs:unique>
    </xs:element>
</xs:schema>
```

Fig. 5. Access Control specification

An obvious omission, from our formal model is that no information concerning service methods is defined within the introduced XML specifications. Apparently, no *role declarations,* as defined in the formal model, exist but roles are directly associated with services instead of methods. Role declarations were deliberately not included in our proposed schemes as there are already related XML specifications, which are widely used today. The most noteworthy of these schemes is the EJB 2.x declarative security specification [10], [11] an example of which is cited in figure 6.



```
<assembly-descriptor>
  <method-permission>
    <role-name>Administrator</role-name>
    <method>
      <ejb-name>PositioningService</ejb-name>
      <method-name>getLocation</method-name>
      <method-params>
        <method-param>java.lang.String</method-param>
      </method-params>
    </method>
  </method-permission>
</assembly-descriptor>
```

Fig. 6. Declarative security definition in EJB 2.0 (excerpt from the EJB deployment descriptor)

4.2 Security mechanisms

A fundamental part of the security framework is the *security context,* which is created after a successful *id* is detected. The security context is an internal memory object indexed by the unique user *id* which holds all security information related to the specific user. An example of its structure (i.e., supported fields) is presented in figure 7. Following its creation, the security context is updated with the appropriate security information for the designated user.

Id	u1235678
System access	OK
Accessible Services	Service1
	Service4
	…
roles:	service1.role1
	service2.role1
	service2.role4
Valid until	12/4/2003 11:52

Fig. 7. Security Context definition

Each entry of the security context is filled with the appropriate information by the corresponding security mechanism. Low-level access control sets the *system access* field, while high-level access control updates the *accessible services* field with all services available to the user. Finally the role-based authorization process retrieves, from the registry, all the available roles for a specific user and inserts them in the *roles* field. The role of the *security context* is to provide some kind of caching mechanism for the information pertaining to the authenticated user in order to speed the authorization and access control process. It can be eliminated without any impact to the pursued functionality, but then each security mechanism will need to consult the Registry for every request submitted to the protected system, even if this request is the same with a previous one. The whole access control process is depicted in figure 8.

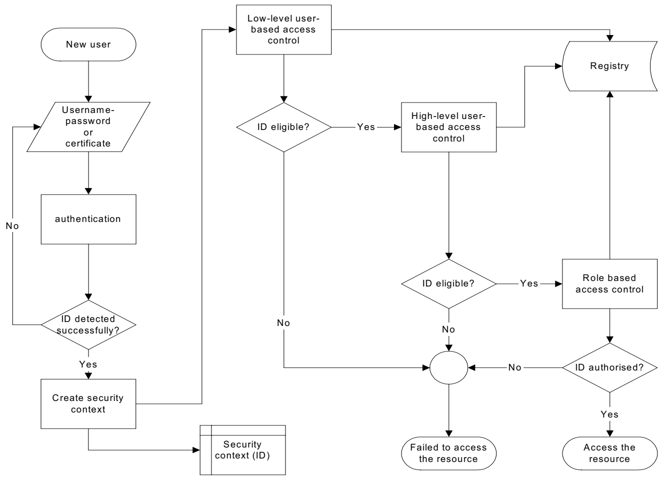

New user
Username-password or certificate
authentication
No
ID detected successfully?
Yes
Create security context
Security context (ID)
Low-level user-based access control
ID eligible?
Yes
No
High-level user-based access control
ID eligible?
Yes
No
Registry
Role based access control
ID authorised?
No
Yes
Failed to access the resource
Access the resource

Fig. 8. Access control process

User-Based Access Control. It includes the low level and the high-level user-based access control mechanisms.

Low-level access control. The low-level access control is the first security mechanism, which can be applied in order to restrict/allow access to the whole protected system. The mechanism is automatically enabled if the accessControl element of the services portion of the Registry is present and set to ENABLED.

Low-level access control, searches all the entries under the aforementioned element for a *member* value that matches the *id* of the user who accesses the system. Depending on whether the *id* is a member of the *allowed* or *notAllowed* element the user can be granted or refused access to the rest of the resources. The mechanism takes also provision for updating the *system access* field of the security context with a Boolean *YES* or NO value. When a stack access control architecture like the one depicted in figure 1 is adopted, further invocation of subsequent access control mechanisms rely on the result produced by this security mechanism.

High-level access control. High-level access control is the second mechanism, which can be enforced. It performs the same operations with the low-level control, which was discussed in the previous section but on the service level this time. Depending on the implementation approach, high-level access control could process the whole Registry (i.e., all service entries) once and update the security context accordingly or perform this check on a per request basis each time access to a new service is requested. Before invoking a certain bundle of resources (e.g. a service), the mechanism checks whether the user is eligible to access the specific service and authorizes his further admission inside the service. Hereafter, the last mechanism (RBAC), undertakes the task of handling the user request.

Role-based Access Control. Supporting different roles per service is the key issue that differentiates the proposed framework from other security infrastructures. The RBAC mechanism performs a two phases process in order to determine if a user is eligible to access a resource inside the multi-service system.

In the first phase the roles of the authenticated user are retrieved and stored inside the *security context*. All *services* inside the Registry are sequentially processed, and if the particular *user* owns a specific role, the corresponding role name is appended in the list of *roles* of the *security context*. A role object consists of the role name, which is specified by the service administrator/creator, prefixed by the service name, thus forming the role *r* as defined in our model. The latter is unique inside the Registry, thus, guaranteeing also the global uniqueness of the role.

The second phase involves the actual access control process. At first, the required roles for accessing the resource/method are determined. This determination could vary depending upon the used *role declaration* scheme. For example, in J2EE environments required the roles could be retrieved, by searching inside the deployment descriptor (see figure 6). Subsequently, the required roles are checked against those present in the *security context*, which were retrieved during the first phase. If a match is found the user is authorized to access the resource. The second phase takes place every time a certain resource is accessed, while the first one only once when the RBAC mechanism is firstly invoked.

5 Conclusions

In this paper, we presented a security framework for controlling access to the critical resources of a computer system. We focused mainly on the definition of the appropriate data structures, which will accommodate the information needed for performing the required security checks. A configurable 3-layer resource access control mechanism, which allows implementation of security mechanism on two levels was also introduced. On the first level a coarse user-based access control is performed on the system's level, while on the second level a fine-grained role-based access control is performed on the service level. The most significant achievement of the framework is that it allows the definition of role names inside a certain service, without influencing other services running on the same computer system; yet each role maintains its uniqueness throughout the whole system, thus allowing the adoption of distributed (i.e., on a service level) role management schemes. The aforementioned characteristic is extremely important in enterprise systems and multi-service environments, as it can significantly reduce the administrative overhead needed for controlling access to their resources.

References

1. D. Ferraiolo, D. R. Kuhn: "Role based access control".,In Proceedings of the 15th Annual Conference on National Computer Security. National Institute of Standards and Technology, Gaithersburg, MD, 554–563,1992.
2. L. Guiri, "A new model for role-based access control", In Proceedings of the 11th Annual Conference on Computer Security Applications (New Orleans, LA, Dec. 1995).
3. L. Guiri, P. Iglio, "A formal model for role-based access control with constraints", In proceedings of 9th IEEE Workshop on Computer Security Foundations, Ireland, 1996.
4. S. Osborn, R. S. Sahdhu, Q. Mutanawer, "Configuring role-based access control to enforce mandatory and discretionary access control policies". ACM Trans. On Information System Security 3, 2 (May 2000).
5. I. Mohammed, D. M. Dilts, "Design for dynamic user-role-based security", Computer Security 13, 8, 661–671, 1994.
6. J. Park, R. Sandhu, G. Ahn, "Role-Based Access Control on the Web", ACM Transactions on Information and Systems Security (TISSEC), Volume 4, Number 1, February 2001.
7. M. Birbeck et al, "Professional XML", Wrox Press Inc, 1st edition, 2000.
8. J. Duckett et al, "Professional XML schemas", Wrox Press Inc, 1st edition, 2001
9. Cattell R. et al, "Java 2 Platform, Enterprise Edition: Platform and Component Specifications", Addison-Wesley Pub Co, 2000.
10. Roman Ed et al., "Mastering Enterprise JavaBeans" 2nd Edition, Wiley Computer Publishing, 2002.
11. Enterprise Java Beans Specification version 2.1,Final Release, Sun Microsystems, 12 November 2003.
12. W. Yao, K. Moody, J. Bacon, "A model of OASIS Role-Based Access Control and its Support for Active Security", proceeding of SACMAT 2001, Chantilly, Virginia, USA, May 3-4, 2001.

On the Role of the Inner State Size in Stream Ciphers

Erik Zenner

University of Mannheim (Germany)
e-mail: zenner@th.informatik.uni-mannheim.de

Abstract. Many modern stream ciphers consist of a keystream generator and an initialisation function. In fielded systems, security of the keystream generator is often based on a large inner state rather than an inherently secure design. As a consequence, an increasing number of attacks on stream ciphers exploit the (re-)initialisation of large inner states by a weak initialisation function.

In this paper, we propose a strict separation of keystream generator and initialisation function in stream cipher design. After giving lower bounds on the necessary inner state size, we show how a secure stream cipher can be constructed from a weak keystram generator. We introduce the notion of inner state size efficiency and compare it for a number of fielded stream ciphers, indicating that a secure cipher can be based on reasonable inner state sizes. Concluding, we ask a number of open questions that may give rise to a new field of research that is concerned with the security of initialisation functions.

Keywords: Stream cipher, keystream generator, initialisation, inner state.

1 Introduction

Background: Let $m = (m_1, m_2, \ldots)$ be a message consisting of blocks $m_t \in \{0,1\}^w$. A stream cipher is a pair of efficient algorithms, where encryption transforms a message block m_t into a ciphertext block $c_t \in \{0,1\}^w$ and decryption implements the inverse transformation. Both encryption and decryption run under the control of a key K and a counter t. Note that the use of a counter is the critical difference between a stream cipher and a block cipher.

A frequent building block for stream ciphers is a keystream generator, i.e. a finite state machine that transforms K into a pseudorandom bitstream $z = (z_1, z_2, \ldots)$ with $z_t \in \{0,1\}^w$. In most cases, z_t is added bitwise to m_t for encryption and to c_t for decryption.

While a large body of literature exists on the design of keystream generators (cf. [33, 27]), the remaining aspects of stream cipher design are less well researched. Only few guidelines exist for the choice of important parameters like key length, inner state size, or the number of bits produced before re-keying. The same uncertainty exists with respect to the key setup algorithm that transforms the key into a starting state for the generator.

The consequences in practical stream cipher design are twofold. On one hand, an increasing number of stream ciphers is broken not by attacking the keystream generator, but by attacking the key setup algorithm (e.g. RC/4 as used in the WEP protocol [35], or

A5/1 from the GSM standard [17]). There exist a few general attack techniques against weak setup functions for stream ciphers (e.g. resynchronisation attacks [13, 23]), but no design criteria for good initialisation functions. Considering recent research progress on related key attacks for pseudorandom functions (see [6] and subsequent work), more problems for stream ciphers designed in an ad-hoc manner are to be expected in the future.

On the other hand, when a cipher is successfully attacked, a common solution is to change the parameters while keeping the general design intact. Examples include increasing the inner state size (e.g. for LILI-128 [9]) or decreasing the security level (e.g. for Sober-128 [25]). For some ciphers, huge security margins for the parameters are used in the first place (e.g. more than 33,000 bit of inner state for SEAL [32]).

Paper outline: This paper intends to be a starting point for future research on the design of stream ciphers. We consider the construction of such ciphers from two primitives: a keystream generator and an initialisation function. Observe that the inner state of the cipher forms the interface between those two primitives. While a large inner state is advantageous for the security of the keystream generator, it makes the task of the initialisation algorithm more difficult. It also requires more memory, hampering the use on a restricted computational device like a smart card. In many respects, inner state bits are to stream ciphers what encryption rounds are to block ciphers: Given a large number of them, almost any design can be secure while at the same time, performance is suffering.

Our goal is to improve the understanding of the necessity and the limitations of the inner state. To this purpose, in section 2, we introduce a formal model of the stream ciphers considered. In section 3, we discuss the cryptographic relevance of inner states, giving lower bounds on the minimum size as well as a construction for a secure stream cipher when inner state size and initialisation time are not critical. In section 4, we observe that the inner state size has to be limited in most practical stream ciphers. This leads us to the definition of inner state efficiency, yielding a measure of how much the inner state size actually contributes to the security of the keystream generator. We also give concrete values of inner state efficiency for a number of practical stream ciphers. Concluding, in section 5, our results are summarised, and a list of open questions for future research on stream cipher design is presented.

2 Terminology

2.1 Keystream generators

Basic model: In [33], Rueppel defined a *keystream generator* as consisting of the following components (see the box in figure 1):

(a) An inner state $S_t \in \mathcal{S}$ with $\mathcal{S} \subseteq \{0,1\}^n$,
(b) an update function $f : \mathcal{S} \rightarrow \mathcal{S}$ that modifies the inner state with each clock, and
(c) an output function $g : \{0,1\}^v \rightarrow \{0,1\}^w$, $w \leq v \leq n$, that uses the inner state to compute w keystream bits with each clock.
(d) a Boolean predicates $C : \mathcal{S} \rightarrow \{0,1\}$, such that an inner state S is a valid starting state iff $C(S) = 1$.

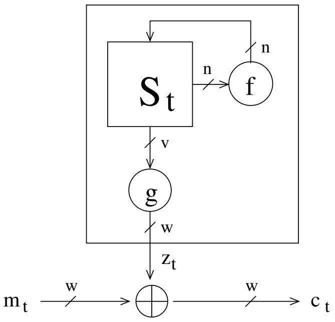


Fig. 1. General model of a keystream generator

Deployment in stream ciphers: Keystream generators are often used in cryptography to implement efficient stream ciphers. A wide-spread design requires the following additional components:

(A) A secret *key* $K \in \{0,1\}^l$ that is not necessarily identical to the inner state,
(B) an *initialisation function* $h : \{0,1\}^l \times \{0,1\}^m \to \mathcal{S}$ that derives a starting state S_0 from the key K and m bit of additional information (like an initial value or a nonce), such that $C(S_0) = 1$.
(C) the *xor-function* $\oplus : \{0,1\}^w \times \{0,1\}^w \to \{0,1\}^w$ which adds the keystream bitwise modulo 2 to the plaintext, generating the ciphertext[1].

2.2 Inner state size:

A naïve candidate for the inner state size is the parameter n, denoting the length of the inner state representation. There is, however, the obvious problem that the same generator may be represented in different ways, yielding different values of n depending on the concrete implementation.

Instead, in order to derive a unique definition of the inner state size, we consider an *autonomous finite state machine* (AFSM) implementing the generator[2]. Such an AFSM consists of a set $\mathcal{S}$ of inner states, and for each inner state $S \in \mathcal{S}$, there exists

- a *transition rule* that defines the next state $f(S)$ for S, and
- a *label* defining the output $g(S)$ generated from S.

In addition, each finite state machine needs a set $\mathcal{S}_0$ of valid *starting states.*

Note that there exists an infinite number of AFSM describing the generator. In particular, the size of the AFSM (i.e. the number of inner states) can vary arbitrarily. Thus, in order to find a unique value for the number of inner states, we need to revert to the notion of the *minimal AFSM* describing the generator.

[1] In fact, other functions like addition modulo 2^w are also possible, but rarely used.

[2] For a full discussion of this topic, cf. [37]

An AFSM is said to *generate* an (infinite) output sequence $z = (z_0, z_1, \ldots)$ if there exists a starting state $S_0 \in \mathcal{S}_0$ such that $z_i = g(f^i(S_0))$. Two AFSM A and B are said to be *equivalent* if all (infinite) output sequences produced by A are also produced by B, and vice versa. As a consequence, all AFSM that describe a given keystream generator are equivalent. An AFSM is said to be *minimal* if no equivalent AFSM of smaller size exists. Thus, if a minimal AFSM for a given generator can be found, its size yields the minimal number of inner states required to implement the generator.

Having introduced these notions, we can define the unique inner state size of the generator as follows:

Definition 1. *Let G be a keystream generator as defined above, and let A be a minimal AFSM implementing G. Then the inner state size of the generator G is defined as* $\hat{n} := \lceil \log_2(|A|) \rceil$, *where* $|A|$ *is the number of inner states of A.*

2.3 Attacker model

Prior knowledge: The attacker is assumed to know everything about the stream cipher with the exception of the key K and the current inner state S_t. In particular, he is aware of the set $\mathcal{S}$ of possible inner states and of the functions f, g and h. He also knows (or even controls) the m bit of public information that are used, along with the key, to set up the inner state S_0.

Class of attack: We consider a known-plaintext attack. Since knowledge of plaintext and ciphertext implies knowledge of the keystream, it can be assumed that the attacker has $L \ll 2^l$ known keystream bits at his disposal.

Computational resources: The attacker can do any computation that requires less steps than a complete search over the key space. This bound holds both for the precomputation and the actual attack phase, implying a bound of strictly less than 2^l bits of memory that can be used.

Notion of success: An attacker is considered successful if he can correctly predict previously unknown keystream bits, or if he can distinguish the output of the keystream generator from truly random bits. We say that the keystream generator is broken if there exists an attacker whose success probability differs significantly from pure guessing. Note that in particular, reconstruction of a correct tupel (t, S_t) or of the key K implies a successful attack in the above sense.

3 Cryptographic strength from large inner states

3.1 The necessity of large inner states

In the following, let l denote the key length, $\hat{n}$ the inner state size and n the length of the inner state representation ($n \geq \hat{n}$). For most practical stream ciphers, it can be observed that $n > l$ holds. We will briefly discuss that this is in fact a necessary condition for secure stream ciphers.

First lower bound: The main design goals of practical stream ciphers are security and efficiency. In order to achieve the efficiency goal, the functions f, g and h are chosen to be as simple as possible. In particular, $g : \{0,1\}^v \rightarrow \{0,1\}^w$ is constructed such that $w \leq v < \min\{l, n\}$.

Lemma 1. *Let the output function g depend on $v < l$ inner state bits and let the output be balanced. Then the keystream generator can not be secure if $n < l + w$.*

Proof. For such a generator, a divide-and-conquer attack can be mounted: The attacker guesses all v bits of the inner state representation that are input to g (since $v < l$, this is feasible in our attack model). He then verifies whether the output of g matches the observed value z_0. Since g is balanced, only 2^{-w} of all assignments meet this criterion, strongly reducing the search space. The attacker can now mount a complete search over the remaining assignments, yielding an attack in 2^{n-w} steps. If $n < l + w$, this attack would be more efficient than brute force search over the key space of the stream cipher. □

Since the value n depends on the implementation and is thus not under the control of the cipher designer, the inner state size must be chosen such that the above attack becomes infeasible for all implementations.

Corollary 1. *If $v < l$, a necessary condition for a secure keystream generator is $\hat{n} \geq l + w$.*

Note that for many ciphers, this attack can be extended using a backtracking approach [21, 38, 36], yielding an even greater lower bound on the minimum size of the inner state.

Second lower bound: The requirement for a large inner state gets even stronger if the attacker has a large amount of keystream bits at his disposal. In this case, time-memory-data tradeoff attacks have to be taken into account, as follows.

Lemma 2. *Let L be the number of keystream bits available to the attacker. Then the keystream generator can not be secure if $n < l + \log(L)$.*

Proof. A general time-memory-data tradeoff [3, 21, 7] for $w = 1$ can be conducted as follows:

- *Precomputation phase:* The attacker draws a large sample (say, $2^{l-\epsilon}$) of inner states at random from $\mathcal{S}$. For each sample state S_i, the generator is run to produce an l-bit output z_i. The tuple (z_i, S_i) is stored in a table, indexed by z_i.
- *Attack phase:* The attacker segments the known output stream into roughly L overlapping frames $\tilde{z}_j$ of l bits[3]. For each frame, he checks whether $\tilde{z}_j$ is contained in the table, and if yes, he extracts the inner state S.

[3] To be exact, there are $L - l + 1$ such frames.

By the birthday paradox, there is high probability for a collision between the set of samples z_i in the table and the set of observations $\tilde{z}_j$ in the keystream if $2^{l-\epsilon} \cdot L \approx 2^n$. Since this attack requires $2^{l-\epsilon}$ precomputations and L table-lookups, it is feasible for the attacker if $n \approx l + \log(L) - \epsilon$, where ϵ is small.
Note that this proof can be generalised for arbitrary values of w by using frame lengths that are multiples of w, yielding the same result. □

Again, the cipher designer can not control n, but only the inner state size $\hat{n}$. Remembering that an attacker who is restricted to 2^l operations can read at most $L = 2^l$ output bits, we obtain the following lower bound:

Corollary 2. *If the generator produces arbitrarily large output streams, a necessary condition for a secure keystream generator is $\hat{n} \geq 2l$.*

3.2 A generic construction

We have seen that for efficient and secure stream ciphers, the inner state size $\hat{n}$ must be strictly larger than the key size l. An obvious question is: What happens if we increase $\hat{n}$ further? An interesting observation is that a large inner state can be used to make up for the deficiencies of a relatively weak keystream generator.

Constructing the stream cipher: Let $H = \{H_n \mid n \in \mathbb{N}\}$ be a family of cryptographically secure hash functions $H_n : \{0,1\}^* \rightarrow \{0,1\}^n$. Let $G = \{G_n \mid n \in \mathbb{N}\}$ be a family of keystream generators with $n = \hat{n}$.[4] Furthermore, let the generator be such that the mapping from state space to the first n keystream bits is bijective. Finally, we assume that there exists a known parameter c, $0 < c < 1$, such that for any generator $G_n \in G$ and given n bits of output stream, predicting additional output bits will require at least 2^{cn} computational steps for all but $O(1)$ cases.

Given these building blocks, we can construct a stream cipher with security level l as follows. First, choose n such that $c \cdot n > l$, and use G_n as keystream generator. The n bits of inner state for generator G_n are initialised using the matching hash function $H_n : \{0,1\}^l \times \{0,1\}^m \rightarrow \{0,1\}^n$.

Security against inversion: It can be shown that such a stream cipher is secure against inversion attacks, as long as no assumption about G_n and H_n is violated.

Lemma 3. *If the stream cipher (G_n, H_n) can be inverted in less than 2^l steps, then the hash function H_n can be inverted in less than $2^l + n$ steps.*

Proof. Assume that there exists an attacker A who, given the description of (G_n, H_n) and at least n bit of cipher output z, can invert the stream cipher in less than 2^l steps. Then we can construct an inverter A' who, given a valid output y of the hash function H_n, finds a corresponding input x such that $H_n(x) = y$.

[4] Many keystream generators are of that kind, e.g. many LFSR-based combination and filter generators or clock-controlled generators.

- A' runs the keystream generator on inner state representation y, producing n bit of cipher output $z = G_n(y)$.
- A' invokes attacker A with input z and obtains a key k with $G_n(H_n(k)) = z$.
- A' outputs k.

Note that k meets the condition $G_n(H_n(k)) = z$. Since G_n is injective, there exists only one intermediate value y with $G_n(y) = z$, implying that $H_n(k) = y$. Thus, A' has inverted the hash function, using $2^l + n$ computational steps. □

Security against prediction: Analogously, it can be shown that the stream cipher is secure against prediction attacks, as long as the output of keystream generator G_n can not be predicted in less than 2^{cn} computational steps in all but a small number of cases.

Lemma 4. *If the stream cipher (G_n, H_n) can be predicted in less than 2^l steps, then the keystream generator G_n can be predicted in less than 2^l steps on a significant subset of its inputs.*

Proof. Assume that there exists an attacker A who, given the description of (G_n, H_n) and output bits $(z_0, \ldots, z_{n-1})$, can predict output bits $(z_n, \ldots, z_{n+d-1})$ correctly in less than 2^l steps. Then we can construct a trivial predictor A' who, given a valid output $(z_0, \ldots, z_{n-1})$ of G_n, can predict the subsequent output bits $(z_n, \ldots, z_{n+d-1})$ in at least 2^l different cases.

- A' runs A on input $(z_0, \ldots, z_{n-1})$ and obtains bits $(z_n, \ldots, z_{n+d-1})$.
- A' outputs $(z_n, \ldots, z_{n+d-1})$.

Note that due to the injectivity of G_n, $(z_0, \ldots, z_{n-1})$ was generated from a unique starting state S_0. For the analysis, we have to distinguish two cases:

(a) If S_0 is a possible output of H_n, the sequence $(z_0, \ldots, z_{n-1})$ is a correct output of the stream cipher (G_n, H_n). Thus, if A predicts correctly for the stream cipher, A' predicts correctly for the generator.
(b) If, however, no key k exists such that $H_n(k) = S_0$, the behaviour of A (and thus of A') is undefined - the prediction may or may not be correct.

In any case, the running time of A' is identical to the running time of A, yielding an effort of less than 2^l steps. Note that the algorithm is always right if case (a) occurs, yielding a correct prediction in at least 2^l (out of 2^n) cases. □

4 Inner state efficiency

4.1 Disadvantages of large inner states

In practice, stream ciphers often use a relatively weak keystream generator and rely on the inner state size and the key schedule algorithm for security. Since constructing a cipher in the above way is tempting, why not use it as a general design rule?

With all their advantages as demonstrated in sections 3.1 and 3.2, there are also three arguments against large inner states. The first (and most obvious) one is that memory is not for free. While on a modern PC, sufficient memory should be available, other platforms like encryption hardware or smartcards may require a more economical use of resources. A second problem is that cryptographic memory must be protected from observation (both on general purpose and specialised hardware), and that an increase in memory size increases the options of an attacker, e.g. for side-channel attacks.

But there is third, more subtle reason why large inner states do not only provide advantages: most practical stream ciphers have to be re-initialised on a regular basis; i.e. after producing a fixed number of output bits, a new inner state is computed from the same key K, but with different additional information. This can be for synchronisation purposes, but also due to cryptographical reasons.[5] However, for a stream cipher with a large inner state, either performance or security of the initialisation procedure is impaired.[6] If the cipher re-initialises rather often, the function h must be computable in as few computational steps as possible. On the other hand, a good mixing of key and nonce into the starting state can not be obtained in a small number of computational steps. The lack of widely accepted design criteria for such initialisation functions further complicates the problem.

4.2 Efficient inner state size

For all of the above reasons, the inner state size must not be too large, even though a certain minimum size for the inner state is necessary, as shown in subsection 3.1. Note that the lower bound on the required inner state size depends on the quality of the keystream generator. In order to make comparisons between different generators possible, we introduce a measure of inner state efficiency, much in analogy to the well-known efficient key size.

Definition 2. *Let G be a keystream generator, and let A be the best known attack against G. The* efficient inner state size *of G is a number $\sigma \in \mathbb{R}$ such that executing A takes as many computational steps as a brute force search over 2^σ starting states of G. The quotient $\gamma = \sigma/\hat{n}$ is denoted as the* inner state efficiency *and is a measure for the quality of the keystream generator G.*

4.3 Comparison of fielded stream ciphers

For an arbitrary generator, the inner state efficiency lies in the range of $0 \leq \gamma \leq 1$. However, using the time-memory-data tradeoff technique presented in subsection 3.1, it can be shown that for a generator that produces at least $2^{\hat{n}/2}$ output bits, the inner state efficiency is restricted to $0 \leq \gamma \leq 0.5$. But what values of γ are encountered in practice?

In table 1, the efficiency δ of inner states is compared for a number of contemporary stream ciphers. We denote by $l_{\max}$ the maximum key length of the overall stream cipher.

[5] Note that once the number of keystream bits available to the attacker gets large, most keystream generators become vulnerable to a wide range of cryptanalytic techniques, like time-memory-data tradeoffs, correlation attacks, differential attacks, or algebraic attacks.

[6] This line of research was first proposed to us by W. Meier [31].

Cipher		$l_{\max}$	$\hat{n}$	σ	γ
A5/1	[8]	64	64	32.0	0.5000
Lili-128	[14]	128	128	48.0	0.3750
E$_0$	[1]	128	132	49.0	0.3712
Sober-t32	[26]	256	576	158.0	0.2743
SNOW 1.0	[15]	256	576	100.0	0.1736
Scream	[24]	128	4,116	100.0	0.0243
RC4 (8bit)	[29]	256	1,700	30.6	0.0180
Leviathan	[30]	256	6,784	39.0	0.0057
Seal 3.0	[32]	160	33,036	43.0	0.0013

Table 1. Keystream generators of fielded stream ciphers (details: appendix A)

Note that σ represents the most efficient attack that (a) has been published at the moment and that (b) targets the keystream generator only. Also note that runtime estimates for cryptographic attacks are always somewhat tricky, but the order of magnitude should be correct. In appendix A, a short description is given on how values $\hat{n}$ and σ have been derived.

It can be observed that the stream ciphers with particularly large internal states have very low inner state efficiencies. It could be argued that the attacks that determined σ for these ciphers are distinguishing attacks, and that in all cases, it is not known how to transform them into prediction or key reconstruction algorithms.[7] But then again, distinguishing attacks could be mounted against the ciphers with smaller inner states, too, with less drastic results. As a consequence, the ciphers with large inner states do not seem to enjoy a real advantage over ciphers with small values for $\hat{n}$. In other words: It should be possible to construct a secure stream cipher from a generator with a reasonable inner state size.

5 Conclusions and research directions

Directions for future research: The following is an incomplete list of open questions that might be of interest for a more thorough understanding of stream cipher design.

- How much inner state is required to make a stream cipher secure?
- What is the right cryptographic primitive for a key schedule algorithm? Is a full-scale pseudorandom function generator required, or can we get away with a less strict security requirement?
- What constructions for provably secure stream ciphers can be given, in particular in the concrete security setting [5]?
- Can a set of practical design guidelines for key schedule algorithms be developed, in analogy to the design guidelines for block ciphers or keystream generators? What

[7] In addition, these ciphers are not meant to be re-synchronised frequently. Thus, they can indeed afford larger internal states under running time considerations, since re-initialisation is required only rarely.

knowledge about key schedule algorithms from block ciphers can be re-used for stream cipher initialisation?
- What are the conditions for a direct attack on the key? What can be said about the relationship between the keystream generator and the key schedule? Is it possible to develop a good stream cipher by using generic keystream generators and key setup algorithms independently? Or should both algorithms be chosen in an orthogonal way, being dependent on one another?

Summary: We have have shown that for a wide-spread type of stream ciphers, an increase in inner state size can increase the security of the keystream generator used, but at the same time can slow down or even weaken the initialisation function. As a consequence, we propose to evaluate the strength of the keystream generator used relative to its inner state size. To this end, the notion of *inner state efficiency* was introduced. In an ad-hoc survey of practical stream ciphers, ciphers with large inner states displayed no cryptographical advantage over those with small inner states. This is an indication that efficient use of inner states is not just a theoretical claim, but is feasible in practice. To this end, we gave a number of questions that may be addressed by future research.

Acknowledgements

The author wishes to thank Frederik Armknecht, Stefan Lucks, Willi Meier, and Greg Rose for encouragement and inspiring discussions.

References

1. *Bluetooth Specification v1.1*, 1999. www.bluetooth.com.
2. F. Armknecht and M. Krause. Algebraic attacks on combiners with memory. In D. Boneh, editor, *Proc. Crypto 2003*, volume 2729 of *LNCS*, pages 162–175. Springer, 2003.
3. S. Babbage. A space/time tradeoff in exhaustive search attacks on stream ciphers. In *European Convention on Security and Detection*, volume 408 of *IEE Conference Publication*, May 1995.
4. S. Babbage, C. De Cannière, J. Lano, B. Preneel, and J. Vandewalle. Cryptanalysis of Sober-t32. In T. Johansson, editor, *Proc. Fast Software Encryption 2003*, volume 2887 of *LNCS*, pages 111–128. Springer, 2003.
5. M. Bellare. Practice-oriented provable security. In I. Damgård, editor, *Lectures on Data Security*, volume 1561 of *LNCS*, pages 1–15. Springer, 1999.
6. E. Biham. New types of cryptanalytic attacks using related keys. In T. Helleseth, editor, *Proc. Eurocrypt '93*, volume 765 of *LNCS*, pages 398–409. Springer, 1993.
7. A. Biryukov and A. Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers. In T. Okamoto, editor, *Proc. Asiacrypt 2000*, volume 1976 of *LNCS*, pages 1–13. Springer, 2000.
8. M. Briceno, I. Goldberg, and D. Wagner. A pedagogical implementation of A5/1. `http://www.scard.org/gsm/a51.html`.
9. A. Clark, E. Dawson, J. Fuller, H.-J. Lee J. Dj. Golić, W. Millan, S.-J. Moon, and L. Simpson. The LILI-II keystream generator. In L. Batten and J. Seberry, editors, *Proc. ACISP 2002*, volume 2384 of *LNCS*, pages 25–39. Springer, 2002.

10. D. Coppersmith, S. Halevi, and C. Jutla. Cryptanalysis of stream ciphers with linear masking. In M. Yung, editor, *Proc. Crypto 2002*, volume 2442 of *LNCS*, pages 515–532. Springer, 2002.
11. N. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In D. Boneh, editor, *Proc. Crypto 2003*, volume 2729 of *LNCS*, pages 176–194. Springer, 2003.
12. P. Crowley and S. Lucks. Bias in the LEVIATHAN stream cipher. In M. Matsui, editor, *Proc. Fast Software Encryption 2001*, volume 2355 of *LNCS*, pages 211–218. Springer, 2002.
13. J. Daemen, R. Govaerts, and J. Vandewalle. Resynchronisation weakness in synchronous stream ciphers. In T. Helleseth, editor, *Proc. Eurocrypt '93*, volume 765 of *LNCS*, pages 159–167. Springer, 1994.
14. E. Dawson, A. Clark, J. Golić, W. Millan, L. Penna, and L. Simpson. The LILI-128 keystream generator. `http://www.isrc.qut.edu.au/resource/lili/lili_nessie_workshop.pdf`.
15. P. Ekdahl and T. Johansson. SNOW - a new stream cipher. `http://www.it.lth.se/cryptology/snow/`. NESSIE project submission.
16. P. Ekdahl and T. Johansson. A new version of the stream cipher SNOW. In H. Heys and K. Nyberg, editors, *Proc. SAC 2002*, volume 2595 of *LNCS*, pages 47–61. Springer, 2002.
17. P. Ekdahl and T. Johansson. Another attack on A5/1. *IEEE Trans. Information Theory*, 49(1):284–289, 2003.
18. H. Finney. An RC4 cycle that can't happen. Newsgroup post to sci.crypt, September 1994.
19. S. Fluhrer. Cryptanalysis of the SEAL 3.0 pseudorandom function family. In M. Matsui, editor, *Proc. Fast Software Encryption 2001*, volume 2355 of *LNCS*, pages 135–143. Springer, 2002.
20. S. Fluhrer and D. McGrew. Statistical analysis of the alleged RC4 keystream generator. In B. Schneier, editor, *Proc. Fast Software Encryption 2000*, volume 1978 of *LNCS*, pages 19–30. Springer, 2001.
21. J. Golić. Cryptanalysis of alleged A5 stream cipher. In W. Fumy, editor, *Proc. Eurocrypt '97*, volume 1233 of *LNCS*, pages 239–255. Springer, 1997.
22. J. Golić. Linear statistical weakness of alleged RC4 keystream generator. In W. Fumy, editor, *Proc. Eurocrypt '97*, volume 1233 of *LNCS*, pages 226–238. Springer, 1997.
23. J. Golić and G. Morgari. On the resynchronization attack. In T. Johansson, editor, *Proc. Fast Software Encryption 2003*, volume 2887 of *LNCS*, pages 100–110. Springer, 2003.
24. S. Halevi, D. Coppersmith, and C. Jutla. Scream: A software-efficient stream cipher. In J. Daemen and V. Rijmen, editors, *Proc. Fast Software Encryption 2002*, volume 2365 of *LNCS*, pages 195–209. Springer, 2002.
25. P. Hawkes and G. Rose. Primitive specification for Sober-128. `http://www.qualcomm.com.au/Sober128.html`.
26. P. Hawkes and G. Rose. Primitive specification and supporting documentation for Sober-t32. NESSIE project submission, October 2000.
27. S. Jiang and G. Gong. Cryptanalysis of stream cipher - a survey. Technical Report CORR-2002-29, University of Waterloo, 2002.
28. T. Johansson and A. Maximov. A linear distinguishing algorithm on Scream. Presented at ISIT 2003, available at `http://www.it.lth.se/movax/Publications/2003/Scream/disting.pdf`.
29. Itsik Mantin. Analysis of the stream cipher RC4. Master's thesis, Weizmann Institute of Science, Rehovot, Israel, November 2001.
30. D. McGrew and S. Fluhrer. The stream cipher Leviathan. NESSIE project submission, October 2000.

31. W. Meier. personal communication, August 2003.
32. P. Rogaway and D. Coppersmith. A software-optimized encryption algorithm. *Journal of Cryptology*, 11(4):273–287, Fall 1998.
33. R. Rueppel. Stream ciphers. In G. Simmons, editor, *Contemporary Cryptology - The Science of Information Integrity*, pages 65–134. IEEE Press, 1992.
34. M.-J. Saarinen. A time-memory tradeoff attack against LILI-128. In J. Daemen and V. Rijmen, editors, *Proc. Fast Software Encryption 2002*, volume 2365 of *LNCS*, pages 231–236. Springer, 2002.
35. A. Stubblefield, J. Ioannidis, and A. Rubin. Using the Fluhrer, Mantin and Shamir attack to break WEP. Technical Report TD-4ZCPZZ, AT&T labs, August 2001.
36. E. Zenner. On the efficiency of clock control guessing. In P. J. Lee and C. H. Lim, editors, *Proc. ICISC '02*, volume 2587 of *LNCS*, pages 200–212. Springer, 2003.
37. E. Zenner. On the role of the inner state size in stream ciphers. Technical Report Informatik TR-04-001, University of Mannheim (Germany), January 2004. available at `http://www.informatik.uni-mannheim.de/techberichte/html/TR-04-001.html`.
38. E. Zenner, M. Krause, and S. Lucks. Improved cryptanalysis of the self-shrinking generator. In V. Varadharajan and Y. Mu, editors, *Proc. ACISP '01*, volume 2119 of *LNCS*, pages 21–35. Springer, 2001.

A Deriving $\hat{n}$ and σ

In the following, the inner states and best known attacks for the keystream generators in section 4.3 are discussed. In most cases, inner states can be subdivided into a linear part (i.e. inner states whose update function is linear), a nonlinear part, and key-dependent S-boxes which may or may not be bijective.

Note that for each stream cipher, only those attacks that target the keystream generator directly are considered.

- *A5/1:* The inner state is purely linear, consisting of 64 bit.
  Numerous attacks have been proposed against the full A5/1 stream cipher, all taking into account that in practice, only a small number of output bits is available to the attacker. If, however, we assume an arbitrary amount of output bits, the generic time-memory-tradeoff attack is most efficient, yielding $\sigma = 32$.
- *E_0:* The inner state consists of a 128-bit linear part and a 4-bit nonlinear part.
  A number of attacks against the E_0 generator have been proposed in literature. For the time being, the algebraic attack technique proposed by Courtois [11] using equations developed by Armknecht and Krause [2] seems to be the most efficient, requiring roughly 2^{49} computational steps. The attack does, however, require more consecutive output bits than the cipher produces between two re-initialisations.
- *Leviathan:* The inner state consists of a 48-bit counter and 4 permutation tables over $\{0,1\}^8$. Thus, the overall inner state size is $16 + 4 \cdot 1,684 = 6,784$ bit.
  The best known attack against Leviathan is a distinguisher by Crowley and Lucks [12], requiring 2^{39} bits of output and a similar work effort.
- *LILI-128:* The inner state consists of two independent linear states of sizes 39 and 89 bit, respectively, yielding a total inner state size of 128 bit.
  Given the construction of the cipher, a security level of 128 bit was not achievable

in the first place due to lemma 1. As a consequence, a number of attacks on LILI-128 have been published, the most efficient one being a specialised time-memory attack by Saarinen [34] that requires roughly 2^{48} computational steps. Note that the attack proposed by Courtois in [11] formally requires less computational steps, but needs 2^{60} output bits.

In the meantime, a successor LILI-II has been published [9]. The inner state size has been almost doubled to 255 bits, with the allowed key size still at 128 bit. No cryptanalysis of LILI-II has been published so far.

- *RC4 (8-bit version):* The inner state size of RC4 is difficult to analyse. It consists of two 8-bit state variables and a table that implements a permutation $\{0,1\}^8$ that changes over time. Normally, this would yield an inner state size of $16 + 1,684 = 1,700$ bit. However, the starting values for the state variables are key-independent, and it was shown by Finney [18] that a fraction of $1/256$ states can never be reached. Experiments on smaller versions of RC4 seem to indicate that the fraction of non-reachable states is even larger but still small enough that $1,700$ is a good approximation of the inner state size.

  Numerous attacks against RC4 have been described. A particularly strong attack against its keystream generator was proposed by Golić [22] and improved by Fluhrer and McGrew [20]. The attack is a distinguisher that requires $2^{30.6}$ output bits and a similar amount of work.
- *Scream:* The generator uses an evolving state of 128 bit, a round key of 256 bit, a mask table (16 times 16 bytes) of 2,048 bit, and an S-Box that implements a permutation over $\{0,1\}^8$ (1,684 bit). Thus, the inner state size is 4,116 bit.

  The best attack against Scream that we are aware of is a linear distinguisher by Johansson and Maximov [28]. It requires about 2^{100} output bits and a similar work effort. Note, however, that Scream has been published only quite recently, i.e. it has not yet received the full cryptanalytic consideration.
- *Seal 3.0:* The generator uses a 12 bit counter, 8 32-bit state words, and a set of lookup tables consisting of 1024 32-bit words, contributing up to $32,768$ bit to the inner state. Thus, the inner state size of the generator is $33,036$ bit.

  While the state words are re-initialised every $2^6 \cdot 2^7 = 2^{13}$ output bits, the tables are re-initialised once every 2^{19} output bits. Thus, SEAL has two initialisation functions h_1 and h_2, and can be seen considered as a stream cipher $(H,G) = ((h_1,h_2),g)$. Note that the best known attack - a distinguishing attack by Fluhrer [19] that requires rougly 2^{43} computational steps - is only applicable if (h_2,g) is considered as the keystream generator.[8]
- *Snow 1.0:* The linear part contributes 16 32-bit words to the inner state, while the nonlinear part adds another 2 32-bit words, yielding a total inner state size of 576 bit.

  Amongst the attacks proposed against Snow 1.0, the most efficient is a distinguisher by Coppersmith et al. [10], requiring about 2^{100} computational steps.

  An updated version Snow 2.0 with equal inner state size has been proposed [16], but no cryptanalytic results are available yet.

[8] The inner generator g is in fact a one-time pad: Without additional knowledge about the initialisation function, it is secure in an information-theoretical sense.

- *Sober-t32:* Here, the inner state consists of 17 linear 32-bit words and one 32-bit constant. Thus, the inner state size is 576 bit.
  The most efficient attack against full Sober-t32 is a distinguisher presented by Babbage et al. [4], requiring $2^{153+5} = 2^{158}$ output bits and a similar work effort.
  Recently, a new version Sober-128 with equal inner state size but reduced key length was published [25]. However, no cryptanalytic results are available for the time being.

An Access Control Model for Geographic Data in an XML-based Framework

Bat-Odon Purevjii[1], Toshiyuki Amagasa[2], Sayaka Imai[1], and Yoshinari Kanamori[1]

[1]Department of Computer Science
Faculty of Engineering, Gunma University
1-5-1 Tenjin-cho, Kiryu, Gunma 376-8515, Japan
{batodon, sayaka, kanamori}@dbms.cs.gunma-u.ac.jp
[2]Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
amagasa@is.aist-nara.ac.jp

Abstract. XML is accepted as a standard format for data exchange on the Web. XML security research has mainly been focused on textual documents since its origin. The previous works are not appropriate in the context of geographic data, which comprises spatial and non-spatial data. Controlling access to those data was a troublesome task because of proprietary and composite data formats of GISs and complex data structures of spatial database systems. Besides allowing flexible integration of geographic data, XML offers us methods to deploy access control for these systems. In this paper we propose an access control model for Web GIS, in which controlled access depends on spatial extents of geographic data, by employing XML, and XML-based 2D vector graphic format SVG. We show its utility in the domain of Internet Mapping into the application of public safety and disaster management for national and local governments.

1 Introduction

Recent Web and XML technologies allow us to transfer from conventional Geographic Information System (GIS) to Web/Human GIS. As a result, every user on the Internet can retrieve geographic information using communication devices, such as PCs, mobile phones, and PDAs, independent of his/her location. The GIS paradigm is shifting from expert-oriented GIS to low-cost consumer oriented GIS which provides an open and rich information container for typical users on the Internet.

Internet Mapping is the major integral part of Web GIS, which can be implemented either within GIS or separately by using XML technologies as glue. The W3C's Scalable Vector Graphics (SVG) [15] has become a new visualization format of such geographic and map data on the Internet[1] [6, 13].

[1] In fact, major GIS vendors started to support the format in their product families.

This work is partially supported by JSPS Grant-in-Aid for Scientific Research (B) No. 15300029.

Such powerful Web GIS and online interactive maps support facilities for collecting, manipulating, sharing, visualizing, and analyzing geo-related information and human related information. Since there is no geographical boundary for accessing such private information, there arises specific security concerns in this context.

Applications for ensuring public safety [7, 14], such as disaster management, homeland security, emergency medical services, crime and terror analysis are emerging and have been under the attention of nations for the last few years. When we face a big disaster, it is important to be able to make quick access to geographic and personal data at any level of government, non-government, and private entities for minimizing damage and saving lives [7, 14]. Thus, offering a large amount of personal information with real world spatial positions and addresses, privacy protection is inevitable in this incorporated environment. Such open and rich information sources may be used by various kinds of users for aggressive purposes.

In this paper we introduce a technique to protect sensitive information in a Web GIS environment by developing an XML-based geographic access control model. In particular, we consider spatial extents and levels-of-details (LoD) of geographic data to regulate access.

The remainder of this paper is structured as follows: Section 2 introduces related work and Section 3 provides a demonstrative scenario that sketches the access control policies and requirements. Section 4 presents a brief introduction to SVG vector format and XML foundations. The underlying XML-based geographic data model is presented in Section 5 and the access control model formulation is introduced in Section 6, respectively. Section 7 describes spatial access control enforcement algorithm. Finally, section 8 gives concluding remarks and future improvements in this area.

2 Related Work

Controlling access to XML documents in the context of text data have received notable efforts and the data security community has proposed several models and systems.

Damiani, et al. [3] and Bertino, et al. [1] have developed access control systems and prototypes in which fine-grained and content-dependent control of XML document fragments are realized. Damiani, et al. [4] presented a model for selectively controlling access to fragments of SVG graphics. Authorization objects are defined by explicit object IDs and implicit conditions on the objects. Kodali, et al. [8] have proposed a model for enforcing control to SMIL movies. S. De Capitani di Vimercati [5] has considered temporal aspects of XML and proposed an authorization model for Temporal XML documents.

Since these previous proposals are considered to be inappropriate in a geographic context, we extend them and propose an XML-based access control model by taking into account the peculiarities of geographic data. Specifically, we consider spatial extents and LoD of XML-based geographic data to regulate access.

Chun, et al. [2] have proposed a novel model for protecting a geo-spatial image database by developing an indexing structure for geo-spatial data. However, they did not consider XML formats and map data.

3 Motivation and Demonstrative Scenario

Within a framework of a disaster management system for local and national governments, various kinds of users, (organized as teams, such as administrative, rescue, medical, research and inhabitants etc.,) interact with the system dynamically depending on their needs and responsibilities[2]. The teams are allowed to access only relevant information to execute their job functions, e.g. a medical team has access to the detailed information of invalids and/or people who need special care during a disaster, such as concrete address, current position of invalids, physical condition, etc. On the other hand, a rescue team has access to detailed information of all houses and buildings within the danger area, such as the number of people, the house condition, pictures and detailed information of surrounding buildings, and personal data of members of households etc.

The members of a team need to be classified into levels depending on their administrative units: from local to national (such as area, district, town, city, prefecture and nation). Furthermore, members in the same team at the same administrative unit level are divided into distinct administrative regions, depending on their locations.

Before introducing requirements for access control in such a working environment, we describe some geographic data concepts briefly. In Figure 1 we can see a map fragment depicting layers of geographic information with an instance of descriptive data of a geographic feature (e.g. a building) in a Web GIS environment.

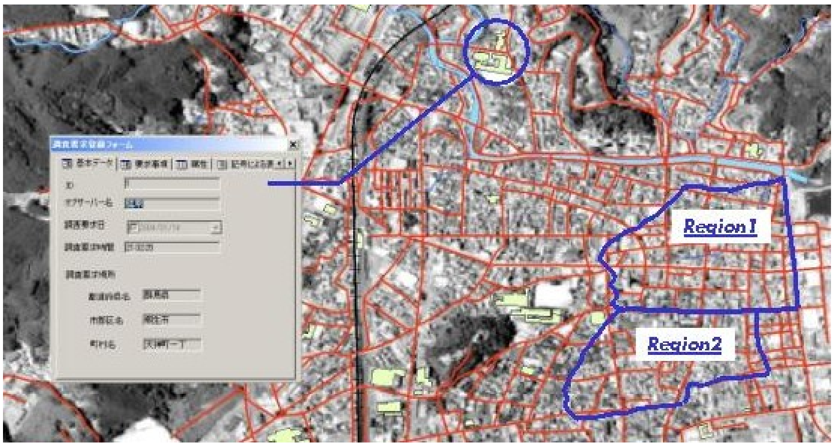


Fig. 1. Fragment of a map in a Web GIS environment.

The descriptive data handles alphanumeric properties of features, and is called the non-spatial component of geographic data. In our case, it can be detailed information of the building. They are stored in a database behind the Website. Besides having descriptions, features have spatial extents, geometry and topology, and are called the

[2] The intention here is to introduce conceivable security requirements in our model but not to illustrate a robust emergency system.

spatial component of geographic data. At the conceptual level, an instance of geographic data is called a geographic object[3] [11].

When a user clicks on a feature, the database query will be issued and related descriptive data is rendered as shown in Figure 1. In a conventional geographic context, users are allowed to access both spatial and non-spatial components of geographic data without any types of security limitations. However, according to security policies in the beginning of this section, only users who belong administratively to *Area1* (on the right side of Figure 1) might be allowed to access geographic objects within this region only, but not in *Area2* or in any of the remaining regions. *Area1* and *Area2* become a bigger administrative region, and the principle should function at this level and for all higher levels up to the national level.

To summarize, the following requirements are derived:

- Flexible organization of various kinds of users by employing any possible combinations of role-based, ID-based and profile-based paradigms;
- Controlling access to geographic objects based on spatial extents;
- Permissions to access geographic data with several LoD;

4 Basic Concepts

4.1 XML basics

To derive a data model of an XML document, graphs and trees are widely employed. XML expresses information using four basic components: elements, attributes, data values, and hierarchy/graph. XML elements are tagged and they contain data values or other elements recursively. Elements may have one or more attributes and the attributes define properties of the elements. In this paper we use a directed graph to structure components of XML because of its expressive power. Vertices of the graph represent elements and attributes, and edges represent relationships between them. We define an XML document as follows:

Definition 1 (XML document). An XML document is a tuple $d=(V^e, V^a, E, Tg, Vl, r, \phi_E')$, where:

- V^e is a set of vertices representing elements,
- V^a is a set of vertices representing attributes,
- E is a set of edges,
- Tg is a set of element and attribute names,
- Vl is a set of element and attribute values,
- r is the root of an XML document,
- ϕ_E is a function that associates edges and vertices.

Here we use an ordered model of an XML document. Each vertex representing an element contains the graph-wide unique identifier.

[3] We use terms geographic data and geographic object interchangeably later on.

4.2 Overview of SVG

Scalable Vector Graphics (SVG) [15] is an XML-based markup language used to describe and integrate two-dimensional vector graphics, raster images and text. Basic geometric/graphics elements are: rectangle, circle, ellipse, line, polyline, polygon, and finally the path object. Graphics in SVG format can be semantically reached and are highly structured. Objects in this structure can be grouped, styled and composed into higher-level objects. SVG offers a number of important advantages over raster formats, especially when it comes to displaying map graphics. The advantages include:

- SVG works well across platforms, output resolutions, and a range of bandwidths
- SVG fully supports the DOM (Document Object Model) and is fully scriptable
- SVG offers greater structural control than other raster and vector graphic formats

5 Geographic Data Model based on XML

The geometric attributes of geographic data are used to effectively store and retrieve such data. In order words, the attributes define only the physical properties of those data. Hence, we need to specify a data model which is capable of expressing geographic objects on a high level before proceeding to the definitions of the access control model.

In GIS, information is organized into layers and a *layer* consists of homogeneous geographic objects/*features* (i.e., objects having the same structure). To model geographic data in XML, we employ XML and XML-based graphic format SVG.

Generally, SVG has five kinds of elements, including graphics/geometric elements, non-rendered text elements, rendered elements by references, reference elements, and container elements. The geometric elements (listed in section 4.2) are the main building blocks of *features* of geographic data. Thus, we assume classifications of SVG elements as only two, geometric and non-geometric, for formal definitions. IDs of *layers*, *features* and SVG elements are unique and can be the graph-wide identifiers or can be defined by the ID attributes of elements. Let *LR* be a set of layers and *FT* a set of *features*.

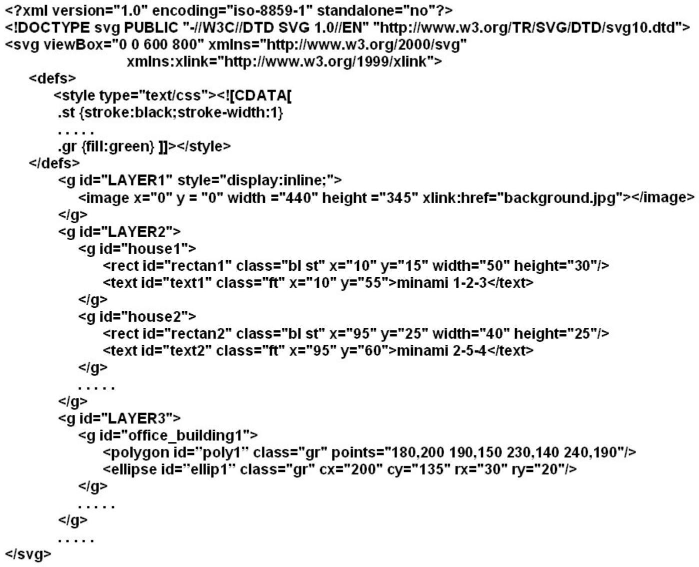
Definition 2 (SVG map). A SVG map is a tuple $sm=(V^e_{sm}, V^a_{sm}, E_{sm}, Tg_{sm}, Vl_{sm}, r_{sm}, \phi_{Esm}, G_{sm})$, where:

- $V^e_{sm} = V^{geo}_{sm} \cup V^{nongeo}_{sm}$ is a set of vertices, where V^{geo}_{sm} are vertices representing geometric elements and V^{nongeo}_{sm} are vertices representing non-geometric elements, respectively.
- V^a_{sm} is a set of vertices representing attributes of all kinds of elements,
- E_{sm} is a set of edges representing relationships among elements;
- Tg_{sm} is a set of element and attribute names,
- Vl_{sm} is a set of element and attribute values.

- r_{sm} is a SVG root element,
- ϕ_{Esm} is a function that associates edges and vertices,

A *feature* can be denoted by listing identifiers of geometric elements in SVG.

Definition 3 (A feature in a SVG map). A *feature* in a SVG map, denoted by *ft*, is a set of identifiers of geometric element in *sm*, that is, $ft=\{id_1, id_2, \ldots , id_n\}$, with $id_i \in \{id_v \mid v \in V_{sm}^{geo}\}$.



```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/SVG/DTD/svg10.dtd">
<svg viewBox="0 0 600 800" xmlns="http://www.w3.org/2000/svg"
                 xmlns:xlink="http://www.w3.org/1999/xlink">
   <defs>
      <style type="text/css"><![CDATA[
       .st {stroke:black;stroke-width:1}
       .....
       .gr {fill:green} ]]></style>
   </defs>
       <g id="LAYER1" style="display:inline;">
          <image x="0" y = "0" width ="440" height ="345" xlink:href="background.jpg"></image>
       </g>
       <g id="LAYER2">
          <g id="house1">
             <rect id="rectan1" class="bl st" x="10" y="15" width="50" height="30"/>
             <text id="text1" class="ft" x="10" y="55">minami 1-2-3</text>
          </g>
          <g id="house2">
             <rect id="rectan2" class="bl st" x="95" y="25" width="40" height="25"/>
             <text id="text2" class="ft" x="95" y="60">minami 2-5-4</text>
          </g>
          .....
       </g>
       <g id="LAYER3">
          <g id="office_building1">
             <polygon id=”poly1” class="gr" points="180,200 190,150 230,140 240,190"/>
             <ellipse id=”ellip1” class="gr" cx="200" cy="135" rx="30" ry="20"/>
          </g>
          .....
       </g>
       .....
</svg>
```

Fig. 2. An example of SVG, presenting a layered map including features.

Consequently, a *layer* can be denoted by listing identifiers of *features* or directly by its own unique ID. A very simple SVG map example is shown in Figure 2.

For definitions of the components of descriptive data of geographic objects, XML formulations are employed. We denote the set of descriptive data as XML description database.

Definition 4 (XML description database). An *XML description database* denoted by *db* is a XML document formulated by Definition 1.

The definitions of *records* and *tables* can be done by listing XML element identifiers of the XML description database. Thus, we denote a set of *records* as *RD* and a set of *tables* as *TB,* respectively.

Definition 5 (Feature-description mapping). Feature-description mapping (depicted in Figure 3) is a one-to-one relation from feature set *FT* to record set *RD*. In order words, there is a function fdm: $FT \rightarrow RD$ where each ordered pair *(a, b)*, $a \in FT$ and $b \in RD$, is unique and is connected by an ID/key.

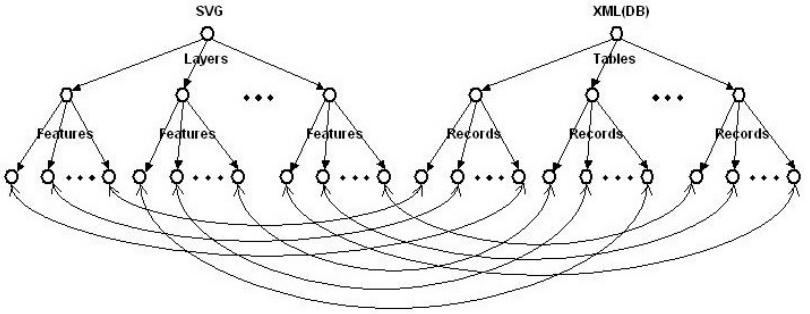


Fig. 3. Feature-description mapping

6 XML-based Geographic Access Control Model

6.1 Authorization object

We define authorization objects directly in terms of XML element identifiers and/or by XPath and XQuery expressions [1, 3] in our model. However, in order not to lose semantics and structural organizations of geographic data, we should define authorization objects by identifiers of features and layers or spatial extents of them. In addition, defining authorizations on each feature is impractical since geographic data hold huge amounts of geographic information that leads the authorizations base to become too large and unorganized. Instead, we define authorization objects at a layer level in authorizations and use spatial extents to filter protection objects within a region in a layer.

6.2 Authorization subject

We employ the integrated approach of user ID, profile and role-based paradigms in our system. Thus, an authorization subject represented as a tuple, (*uid, profile, role-set*), where *uid* is the unique ID of a user, *profile* is a set of user properties, and *role-set* is a set of role names. *Profile* in turn is a tuple (*name, address, work*), where *name* is a user name, *address* is an affiliation address and *work* is an affiliation function, respectively. The users can be modeled and organized by role hierarchies like any other role-based models. In our current proposal we utilize a role-graph structure such as [9] and will develop more flexible structures in further works.

Definition 6. (Authorization subject) An authorization subject specification is the form, structured as XML document:

```
<ELEMENT  subject          (uid, profile, role+)>
<ELEMENT> uid              (#PCDATA)
<ELEMENT> role             (#PCDATA)
<ELEMENT  profile          (name, address, work)>
```

```
<ELEMENT> name              (#PCDATA)
<ELEMENT> address           (#PCDATA)
<ELEMENT> work              (#PCDATA)
```

6.3 Authorization

An authorization in the model is specified as follows:

Definition 7 (Authorization). An authorization a is a tuple (*auth-subj, auth-obj, oper_reg, LoD, md*), where: *auth-subj* is the authorization subject specification as defined in Definition 6, *auth-obj* is the authorization object specification defined as layers, *md* is a set of action modes like view, insert, update, delete, all, etc., *oper_reg* and *LoD* are the values which define a set of operative regions and *LoD* the subject allowed to access. Currently, the *LoD* is defined only on the descriptive database part of objects and the values of *LoD* range from 3 to 1, i.e. from fine-detailed to coarse-detailed.

Example 1: A_1 = ((*su123, null, adm)*, *{*$layer_k$, $layer_{k+1}$, $layer_{k+4}$*}*, *regionL2-1, 1, {all}*): This authorization allows the subject, who is included in role *administrative* and *uid = su123* to execute all action modes (view, insert, update and delete) on objects inside *regionL2-1* on $layer_k$, $layer_{k+1}$ and $layer_{k+4}$ with detail level 1.

Example 2: A_2 = ((*su456, null, medical)*, *{*$layer_{k+1}$, $layer_{k+2}$*}, regionL1-1, 2, {view})*: This authorization allows the subject, who is included in role *medical* and *uid = su456* to execute view mode on objects inside *regionL1-1* on $layer_{k+1}$ and $layer_{k+2}$ with detail level 2.

In the following section we introduce a spatial access control enforcement algorithm.

7 Spatial Access Control Enforcement

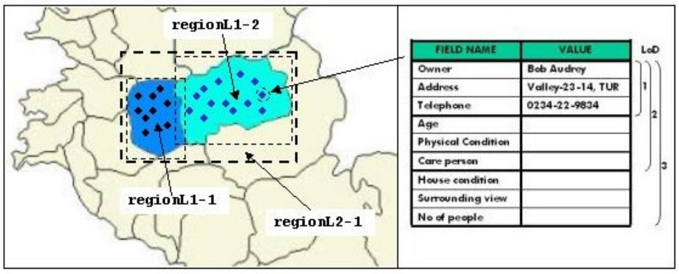
Although access control enforcement is quite similar in many systems [12], in order to manage access control spatially a spatial indexing structure on authorization objects is needed [2]. Chun, et al. [2] have chosen a variant of quadtree, which belongs to the space-driven approach and the main memory access method. To manage access control spatially and effectively for geographic data, however, we need a secondary memory structure and the data-driven approach with LoD support [10]. Since, typically, geographic databases occupy several gigabytes of storage and the distribution of geographic objects on a plane is rambling. Due to space limitation, we will present such an indexing structure in another paper.

Here, we describe spatial access control enforcement in an easier way by employing R-tree based indexes, such as R*-tree. Data values of x, y coordinates of SVG geometric elements express positions of features in a SVG map. Consider features in the SVG map as indexed by R-tree on those values and the leaf nodes of the tree contain IDs of features of the SVG map.

To describe our algorithm we have used literals, because our intention is to give a clear explanation rather to be efficient or provide well-formalized listing. The enforcement algorithm comprises the following main stages:

Step 1. *Authorizations evaluation*: Determine the set of suitable authorizations for the user's request.
Step 2. *Spatially authorized objects calculation*. Retrieve IDs of features by traversing R-tree indices of authorized layers by window queries on the operative region set values of the requestor.
Step 3. *Elimination of unnecessary IDs*. By employing a complex elimination algorithm or using an address-matching function, check and eliminate IDs of features, which do not reside in authorized administrative regions of the requestor.
Step 4. *Retrieval of corresponding records*. By the extracted IDs, parse and retrieve corresponding records with appropriate LoD from the *XML description database*. Retrieval of records with appropriate LoD can be done by XQuery expressions.
Step 5. *Document rendering*. Form a final view of the SVG map only with spatially authorized features and render for the requestor.

During navigation with the map, the requestor can access/see only authorized features. The descriptive data of the features will be depicted on the allowed LoD as well. In Figure 4 we can see an instance of a user (who is allowed to access features in *regionL2-1* with LoD 1) view, which rendered after considering corresponding authorizations.



FIELD NAME	VALUE
Owner	Bob Audrey
Address	Valley-23-14, TUR
Telephone	0234-22-9834
Age	
Physical Condition	
Care person	
House condition	
Surrounding view	
No of people	

Fig. 4. An instance of a user view.

8 Conclusions and Future Work

In this paper we have presented an XML-based access control model for regulating access to geographic data considering spatial extents and LoD. We have shown its utility in the domain of Internet Mapping into the application of a disaster management framework for national and local governments.

Since the work is derived from our preliminary results, a number of improvements and investigations need to be done. The performance issues will be evaluated and efficiency improvements will be conducted. Temporal constraints of access control will also be integrated into the model.

References

1. E. Bertino and E. Ferrari, “Secure and selective dissemination of XML documents”, *ACM Transactions on Information and System Security (TISSEC)*, Vol. 5, No. 3, pp. 290-331, August 2002.
2. S. A. Chun and V. Atluri, “Protecting Privacy from Continuous High-resolution Sattellite Surveillance”, *Data and Application Security: Developments and Directions,* (eds.) Bhavani Thuraisingham, R. Van de Riet, K. R. Dittrich and Z. Tari, Kluwer Academic Publishers, 2001.
3. E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati, “A Fine-Grained Access Control System for XML Documents” *ACM Transactions on Information and System Security (TISSEC)*, Vol. 5, No. 2, pp. 169-202, May 2002.
4. E. Damiani, S. De Capitani di Vimercati, E. Fernandez-Medina, and P. Samarati, “An Access Control Sytem for SVG Documents”, *16-th IFIP Conference on Data and Application Security*, University of Cambridge, UK, July 2002.
5. S. De Capitani di Vimercati, “An Authorization Model for Temporal XML Documents”**,** *Proc. of the 17th ACM Symposium on Applied Computing,* Madrid, Spain, March 2002.
6. R. George, " GIS meets XML: SVG - Scalable Vector Graphics", http://www.web-maps.com/svg-gis.html, 2001.
7. S. Kakumoto, Y.Kosugi, M.Hatayama and H.Kameda, “Development of Spatial Temporal Geographic Information System”, *Technical Report of the Geographical Survey Institute* / 1-No.275-2, March 2002.
8. N. Kodali and D.Wijeseker, “Regulating Access to SMIL formatted Pay-per-view Movies”, *ACM Workshop on XML Security*, George Mason University, Fairfax VA, USA November 2002.
9. S. Osborn and Y. Guo, “Modeling Users in Role-Based Access Control” *Fifth ACM Workshop on Role-Based Access Control*, Berlin, July 2000.
10. P. van Oosterom. “The Reactive-tree: A Storage Structure for a Seamless, Scaleless Geographic Database”, *In proceedings Auto-Carto 10*, Baltimore, March 1991.
11. Ph. Rigaux, M. Scholl, and A. Voisard, “Spatial Databases - with applications to GIS”, Morgan Kaufmann, 2002.
12. P. Samarati and S. De Capitani di Vimercati, "Access Control: Policies, Models, and Mechanisms" in *Foundations of Security Analysis and Design*, R. Focardi and R. Gorrieri (eds), LNCS 2171, Springer-Verlag 2001.
13. DBx Geomatics, SVG mapping, http://www.dbxgeomatics.com/svg.asp
14. FGDC “Homeland Security & GIS” http://www.fgdc.gov/publications/homeland.htm
15. W3C, Scalable Vector Graphics, http://www.w3.org/Graphics/SVG/

Certificate Revocation Lists or Online Mechanisms[1]

Vipul Goyal

Department of Computer Science & Engineering
Institute of Technology
Banaras Hindu University
Varanasi, India
vipulg@cpan.org

Abstract. With more and more acceptance of Digital Certificates and Public Key Infrastructures (PKI), the mechanisms to revoke a certificate in a PKI have recently received increasing attention. The revocation mechanisms are commonly classified into Certificate Revocation Lists (CRLs), trusted dictionaries and online mechanisms. The designer of a PKI should select an appropriate revocation method suiting his requirements. This turns out to be a sufficiently confusing task as different revocation solutions are good in different type of environments. We ask the question "How do we decide which revocation solution to use amongst the various categories of solutions?" We first conduct a survey of the existing certificate revocation techniques and then analyze and compare the various classes of revocation methods for their advantages and disadvantages. This analysis can greatly help the PKI designer to select the right revocation solution.

1 Introduction

A certificate is a digitally signed statement binding the key holder's (principal's) name to a public key and various other attributes. The signer (or the issuer) is commonly called a certificate authority (CA). Certificates act as a mean to provide trusted information about the CA's declaration w. r. t. the principal. The declaration may be of the form-

"We, the Certificate Authority declare that we know Alice. The public key of Alice is ..."

"We further declare that we trust Alice for ..." (optional part)

Certificates are tamper evident (modifying the data makes the signature invalid), unforgeable (only the holder of the secret, signing key can produce the signature). Certificates are the building blocks of a Public Key Infrastructure (PKI). PKI is defined to be "The set of hardware, software people, policies and procedures needed

[1] Throughout the paper, we use examples involving Alice and Bob, where Alice is assumed to be the sender and the subject of the certificate and Bob is assumed to be the acceptor (or the verifier) of the certificate.

to create, manage, store, distribute, and revoke public key certificates based on public key cryptography" in the IETF PKIX Roadmap [1].

When a certificate is issued, the CA (issuer) declares the period of time for which the certificate is valid. However, there may be some situations when the certificate must abnormally be declared invalid prior to its expiration date. This is called certificate revocation. This can be viewed as "blacklisting" the certificate. This means that the existence of a certificate is a necessary but not sufficient evidence for its validity. A method for revoking certificates and distributing this revocation information to all the involved parties is thus a requirement in PKI. The reasons for revoking a certificate may be: suspected/detected key compromise, change of principal name, change of relationship between a principal and the CA (e.g. Alice may leave or be fired from the company) or end of CA's trust into the principle due to any possible reason.

The revocation mechanism should have an acceptable degree of timeliness, i.e., the interval between when the CA made a record of revocation and when this information became available to the relying parties should be small enough to be acceptable. Further, it is very important for the revocation mechanism to be efficient as the running expenses of a PKI derives mainly from administering revocation [4].

2 Available Revocation Techniques

This section briefly outlines a number of available revocation schemes-

2.1 Certificate Revocation Lists (CRLs)

CRLs are the most common and simplest method for certificate revocation. A CRL is a periodically issued list containing the certificate serial number of all the revoked certificates issued by a particular CA. This list is digitally signed by the CRL issuer to avoid tampering. The relying parties willing to validate a certificate issued by a particular CA can then download the most recent CRL of that CA.

Many variants of this "basic" CRL scheme have been designed to improve the performance. These include delta CRLs [2], partitioned CRLs and over-issued CRLs [3].

CRLs have been criticized for not being able to provide the required service and for being too costly [7, 8, 9, 11, 12]. We analyze the arguments advising against and for the use of CRLs in section 3.

2.2 Trusted Dictionaries

There are a number of schemes in which the end entities (relying parties) are supplied information in support of validating a single certificate rather than a complete list. A problem with this method is that the process of digitally signing each revocation reply is processing intensive. Trusted dictionary schemes attempt to solve this problem by using one-way hash functions in order to provide lightweight digital signatures. These techniques include Certificate Revocation Status Directory (CRS Directory) [7], Hierarchical Certificate Revocation Scheme [16], Certificate Revocation Trees [17], Naor's and Nissim's Scheme [10].

Common for most of these schemes are that they are not standardized like CRL and online schemes, and only CRT has been implemented for common use [18]. These schemes do not (with the exception of CRT) support the expressiveness found in CRLs. Additionally, these schemes are difficult to understand and implement as compared to CRLs and online mechanisms. These factors have limited the widespread use of trusted dictionaries.

2.3 Online Revocation Mechanisms

As a response to the low timeliness of some periodically updated certificate revocation schemes, protocols for online status checking have been developed. Many certificate based systems cannot tolerate the revocation delay resulting from the periodically updated schemes. With real time revocation checking, any party can confirm/obtain the proof of the certificate validity by performing an online transaction that indicates the current revocation status for a certificate.

We briefly summarize the common online revocation techniques-

2.3.1 On-Line Certificate Status Protocol (OCSP)

The OCSP [5] is a protocol developed by IETF in which on-line revocation information is available from an OCSP responder thorough a request/response mechanism. OCSP is designed to check the certificate revocation status exclusively.

The protocol is applied between a client (OCSP requester, acting for the user) and a server (OCSP responder, representing a directory). The client generates a so called OCSP request that primary contains one or even more identifiers of certificates queried for validity check, i.e. their serial number together with other data. Then, the (optionally signed) request is send to the server. The server receiving the OCSP request creates an OCSP response: The response mainly includes a timestamp representing the time when the actual request was generated, furthermore, the identifiers and status values of the requested certificates together with a validity interval. A certificate status value is either set to good, revoked or unknown. Be aware that "good" implies three meanings: firstly, the certificate is not revoked, but secondly, it may also not be issued yet or even thirdly, the time at which the response is produced is not within the validity of the certificate. Status "revoked" stands for a revocation or on hold of the certificate. If the answer is "unknown" the server has no information available about the required certificate. The validity interval specifies the time at which the status being indicated is known to be correct and optional the time at or before newer information will be available about the status of the certificate. The OCSP response should be digitally signed either by the server or by the CA. In case of any error the OCSP response contains an error message. The OCSP response is send to the requesting client of the user who then analyzes the data.

Depending on proper defined time schedules, OCSP provides more timely status information than any other method. A pre-producing of signed responses is currently optional. OCSP is especially appropriated for attribute certificates where status information always needs to be up-to-date.

2.3.2 OCSP-X, SCVP and DC

There are a number of on-line protocols that are more extensive than OCSP. OCSP-X [19], or OCSP extensions, provide a richer set of semantics for OCSP. With these extensions, an end entity is able to delegate the full task of deciding whether a certificate should be relied upon and whether it is acceptable for a particular operation.

The Simple Certificate Verification Protocol (SCVP) [20] is a separate protocol that is capable of handling (parts of or) the entire certificate validation process. With SCVP, end entities can avoid the overhead involved in processing the certificate validation locally. The protocol may also be used to centrally enforce some validation policy.

The Data Certification Server (DCS) [21] is a trusted third party that can be used as a component in building non-repudiation services. DCS is capable of verifying the correctness of specific data submitted to it. This service may, for example, be used to verify the correctness of a signature, the full certification path, and the revocation status of a certificate. Note that DCS provides more general services than OCSP-X and SCVP.

2.3.3 Obtaining new certificates

Ronald Rivest [11] criticizes CRLs and points out several design principles which cannot be fulfilled by CRLs. Rivest proposes an online approach in which if the most recent certificate fails to satisfy the recency requirements of the acceptor, the principal should obtain a more recently issued certificate from the CA. Hence, if Alice has a week old certificate indicating employment at the company and Bob is willing to accept at most a day old evidence of employment, Alice should query the online CA and get a new recent certificate created for her. Note that Alice may use this certificate again for other transactions.

The approach clearly has advantages i.e. the acceptor is able to set the recency requirements, certificate validation is reduced to just validating the digital signature on the certificate, acceptor need not deal with any revocation mechanism and better load distribution on the sender and the acceptor. A drawback is the increased load on the certificate servers. The certificate servers are now required to sign many more certificates than before.

3 Certificate Revocation Lists or Online Mechanisms?

While the approach of Trusted Dictionaries has not been deployed in common practice, choosing between CRLs and online mechanisms is still a sufficiently confusing task for PKI designers. Both CRLs and online mechanisms have their own advantages and disadvantages.

It has been argued [7, 8, 9, 11, and 12] at length that CRLs are both semantically and technically inferior to online mechanisms. There are a number of quite convincing arguments supporting this statement. We analyze the reasons from various paper as well as present some new reasons for the criticize of CRLs in the following propositions-

1. Recency Requirement must be set by the acceptor, not by the certificate issuer (CA). The reason is that the acceptor is the party who is running the risk if his decision is wrong, not the CA. Bob may want at most a day old evidence of employment at the bank before granting Alice the access to bank accounts of the customers. Weekly issued CRLs cannot meet his requirements. CRLs require the verifier to accept a recency guarantee bounded by the rate at which CRLs are generated.

2. The cost of CRL management and distribution is too high. Because of the potential size of CRLs, scaling to large communities can be difficult. To verify the certificate of Alice, Bob should download the complete CRL of the Alice's CA. The result of a recent simulation study [18] indicates that the maximum network load in case of CRLs is about 10 times higher than in case of online approaches.

3. CRLs are inappropriate for transactions that require real time revocation state information. That is, the inherent costs of CRLs generation and especially distribution prohibit online CRL generation.

4. For efficiency, the principal (sender) should supply all relevant validity evidence including recency information. More precisely, this states- *"For best load distribution, do work for your certificates yourself"*. There are several reasons for this proposition: - a) the sender can query the CA as well as the acceptor can, b) the recency information obtained may be useful again to the sender, c) this structure puts any burden on the sender (usually the client) rather than on a possibly overworked acceptor (the server). Even in cases, when the sender is the server (e.g. in https protocol, while establish an SSL connection, server sends its certificate), it is not much work for the server to query the CA and obtain a recent certificate daily (or even hourly). This approach is clearly better than having each client obtain the CRL of the server's CA to verify the server's certificate, d) in many case, this allows the acceptor (server) to be implemented in a stateless manner. For example, Bob can reply to Alice, "Sorry, please make sure that your evidences are at most one week old," and then forget about Alice until she comes back again, rather than having to rummage all over the Internet to see if Alice's certificates are still OK. A stateless server design is less vulnerable to denial-of-service attacks.

5. The distribution of request rates for the CRL distribution server is poor. If the weekly CRL is issued by the CA on Monday morning, clearly the request rate for CRLs will be much higher on Mondays and Tuesdays and will be quite low on Saturdays and Sundays. This high peak request rate shoots up the processing and network bandwidth requirements for the CRL server. The requests in case of online mechanisms are perfectly evenly distributed making them the most cost effective solution.
6. New certificates are the best evidence of recency. If a (new) certificate with a guaranteed validity period is available, then the acceptance process may be reduced to the validation of a single certificate signature. As the revocation state is implied by the existence of the certificate, CRLs are unnecessary.
7. Certificates in traditional CRL based schemes do not have any inherent recency information other than the certificate lifetime. Thus, each time a certificate is accessed, the verifier is required to obtain and validate a suitably recent CRL. Combined with proposition 7, this makes a strong argument for the use of online revocation mechanisms [5].
8. CRLs do not provide positive statements. Because CRLs only identify revoked certificates, the existence of a (non-revoked) certificate cannot be determined solely from the validity information.
9. Sometimes, downloading the CRL may introduce unacceptable latency in certificate validation. Since the acceptor should first download the most recent copy of the CRL of the sender's CA before validation (in case it doesn't have one), the delay introduced in the certificate validation may be significant.

These propositions give evidence of the problem with CRL based techniques and argue that CRLs should be eliminated in favor of online mechanisms.

While these arguments are definitely true and convincing, it should be noted that even after having of so many limitation and drawbacks, there exist some PKI environments where CRLs may still be the most cost effective revocation solution. [6] This is because of the two main reasons-

1. Though the bandwidth requirement for CRLs is clearly much higher than that for the online approaches, CRL based mechanisms avoid much of the cost associated with signature generation. Only one digital signature is periodically required by the CRL server. In contrast, online mechanisms place a heavy burden on the revocation server. This demonstrates a chief performance tradeoff between online and CRL mechanisms; CPU cost vs. bandwidth cost.

2. CRL are an attractive option in tightly coupled environments when reference locality is observed, i.e. when the acceptor has to validate many certificates issued by a single CA, periodically downloading the CRL of that CA may be a cheaper option.

A classical example when CRL are usually the best option is the Intranet Service. Certificate in a PKI running on an organization’s Intranet are usually issued by a single/small number of CA's. Hence the relying (accepting) parties have the advantage of reference locality. Further, the bandwidth is not much problem as far as Intranet of any organization is concerned. Hence in this case CRLs are an attractive choice as far as real time revocation is not required.

So, we see that one must select the right PKI solution keeping in mind these points and the availability of resources. If real time revocation is required, online mechanisms are the obvious choice. For others, a careful choice should be made between CRLs and online approaches. The above points may serve as guidelines while selecting the revocation solution.

4 Conclusions

We briefly study the currently available revocation methods. Selecting the right revocation solution is important as the running expenses of a PKI derives mainly from administering revocation. While the approach of trusted dictionaries is limited, CRLs and online methods are commonly used as revocation methods. We study and compare these two approaches in light of their advantages and drawbacks in different environments. We conclude that online mechanisms are generally the most efficient vehicle for the distribution of the revocation information though CRLs should definitely be considered when the PKI environment offers reference locality and does not have bandwidth bottlenecks.

In past, most of the research has focused on creating new and more efficient revocation mechanisms. Now that there are a number of revocation options available, the problem of selecting the right revocation solution for the target environment assumes special importance. In this paper, we provide an analysis of various revocation options resulting in guidelines which one should keep in mind while selecting a revocation solution. These guidelines can greatly ease the task of a PKI designer as far as selecting the right revocation option is concerned.

References

1. A. Arsenault and S. Turner, PKIX Roadmap, Internet Draft, "Work in progress, IETF PKIX working group", October 1999.
2. Warwick Ford and Michel S. Baum, Secure Electronic Commerce, Prentice Hall PTR, 1997.
3. David A. Copper, A model of certificate revocation, proceedings of the Fifteenth Annual Computer Security Application Conference, December 1999.
4. Stuart Stubblebine, Recent-secure authentication: Enforcing revocation in distributed systems, In Proceedings 1995 IEEE Symposium on Research in Security and Privacy, pages 224-234, May 1995.

5. A. Malpani S. Galperin M. Myers, R. Ankney and C. Adams, RFC 2560: X.509 internet public key infrastructure online certificate status protocol - OCSP, June 1999.
6. Patrick McDaniel and Aviel D. Rubin, A response to "can we eliminate certificate revocation lists?", Financial Cryptography, pages 245-258, 2000.
7. S. Micali, Eficient certificate revocation, Technical Memo MIT/LCS/TM-542b, Massachusetts Institute of Technology, Laboratory for Computer Science, March 1996.
8. J. Millen and R. Wright, Certificate revocation the responsible way, Post-proceedings of Computer Security, Dependability and Assurance: from Needs to Solutions (CSDA'98), IEEE Computer Society.
9. M. Myers, Revocation: Options and challenges, Lecture Notes in Computer Science, volume 1465, pages 165-171, 1998.
10. Moni Naor and Kobbi Nissim, Certificate revocation and certificate update, Proceedings 7th USENIX Security Symposium (San Antonio, Texas), Jan 1998.
11. Ronald L. Rivest, Can we eliminate certificate revocations lists? Financial Cryptography, pages 178-183, 1998.
12. Fox and LaMacchia, Certificate revocation: Mechanics and meaning, Financial Cryptography, LNCS, Springer-Verlag, 1998.
13. Carl A. Gunter and Trevor Jim, Generalized certificate revocation, Symposium on Principles of Programming Languages, pages 316-329, 2000.
14. R. Housley, W. Ford, W. Polk, and D. Solo, RFC 2459: Internet X.509 public key infrastructure certificate and CRL profile, January 1999. Status: PROPOSED STANDARD.
15. P. C. Kocher, On certificate revocation and validation, Financial Cryptography, LNCS, Springer-Verlag, 1998.
16. William Aiello, Sachin Lodha, and Rafail Ostrovsky, Fast Digital Identity Revocation, Advances in Cryptology - CRYPTO '98, Springer, 1998.
17. Paul Kocher, A Quick Introduction to Certificate Revocation Trees (CRTs), Technical report, ValiCert, 1999.
18. Andre Arnes, Public Key Certificate Revocation Schemes, Master's thesis, Department of Telematics, Norwegian University of Science and Technology, February 2000.
19. Phillip Hallam-Baker, OCSP Extensions, Internet Draft, "Work in progress, IETF PKIX working group", September 1999.
20. Ambarish Malpani and Paul Hoffman, Simple Certificate Validation Protocol, Internet Draft, "Work in progress, IETF PKIX working group", April 1999.
21. Carlisle Adams and Robert Zuccherato, Data Certification Server Protocols, Internet Draft, "Work in progress, IETF PKIX working group", September 1999.

Semantic Interoperability of Authorizations

Mariemma I. Yagüe, Antonio Maña, Francisco Sánchez

Computer Science Dept. Univ. of Malaga.
E.T.S.I.Informatica. Campus de Teatinos.
29071 Malaga, Spain
{yague, amg, cid}@crypto.lcc.uma.es

Abstract. The shift from paper documents to their respective electronic formats is producing important advantages in the functioning of businesses and Public Administrations. However, this shift is often limited to the internal operation of each entity because of the lack of security in the electronic communication mechanisms. Traditionally, these entities have managed their Local Area Networks (LANs) or even Virtual Private Networks (VPN) as isolated islands, where local identity-based authorization schemes were appropriate. But, the trend towards paperless procedures leads to the need for these entities to interoperate. As an advance, extranets were proposed to connect entities that share common goals in a way that automates their administrative interactions using Internet technology. However, the limited authorization and access control capabilities provided by extranets is a mayor drawback for their application in open and heterogeneous scenarios. Trust appears as the main issue to address in order to achieve secure interoperation of different independent entities. This paper presents a solution to this problem, based on the use of Privilege Management Infrastructures (PMIs) and the semantic description of the different authorization entities.

1 Introduction

Today, being able to procure and provide access to information is a defining characteristic of successful companies. And as companies open up their networks to partners and other third-party users to share information, security has become more important than ever. Companies require comprehensive security systems that allow controlled access.

Therefore, entities need to be able to limit access so that only permitted users have access to certain resources. This means that traditional encrypted tunnels such as VPNs, which require that everyone at both ends is trusted, are inadequate for third-party access. When it comes to sharing information with outsiders, companies need to provide one-way directed access to shared information.

An Extranet is a communication network connecting entities that share common goals in a way that automates their administrative interactions. When properly designed and implemented, extranet systems can be highly effective in improving cross-entity information flows. Extranet services use existing Internet infrastructure, which makes extranets far more economical than proprietary networks. However, the limited authorization and access control capabilities provided by extranets is a mayor drawback for their application in open and heterogeneous scenarios.

Trust appears as the main issue to address in the design of a platform allowing secure interoperation of different independent entities. Many distributed application scenarios such as e-commerce, e-business, e-government, grid computing or web services can benefit from the services of such platform. Some important characteristics of these scenarios are:

- Independence of Authorities. The authorities, as well as the rules governing the functioning of each party are usually predefined and must be independent of others and under control of the legitimate authority.
- Attribute-based access. Usually, access is offered to previously unknown users (individual citizens and members of other entities). Knowledge of their identities, provided by a Public Key Infrastructure (PKI) is not sufficient in order to interact with them.
- Heterogeneity. In open distributed systems we deal with a large number of stakeholders or owners of resources with very different policies and interests, but also with a large number of previously unknown clients, with very different profiles and interests. Moreover, resources under control are intrinsically heterogeneous in type, format, origin, validity, etc. Consequently, the security requirements and access control criteria are also very disparate. As a result, it is impossible for administrators to foresee a fixed role-based structure of the users.
- Flexibility. A high degree of flexibility is required because of the heterogeneous nature of the resources (data and services), access criteria and users. In fact, flexibility appears as one of the most important goals to achieve. The model must be flexible enough to be applicable in different scenarios with few or no changes.
- Scalability. The scalability of the scheme is very important. Therefore, a fully distributed scheme is mandatory. Furthermore, due to the large amount of resources, it is important to be able to determine access conditions automatically, based on their associated semantic information.
- Interoperability. In these scenarios, it is not possible to predict the interactions with other parties. Typically, these interactions will take place only occasionally and parties will frequently be related by a few transactions in common. Because we are dealing with security-sensitive systems, it is essential to guarantee that the interoperation with other parties does not introduce any security weakness.
- Dynamism. This characteristic is essential in most of our targeted scenarios, where the existence of highly dynamic resources is frequent. The access control model must be capable of adapting to frequent changes in access control criteria, client attributes, environment conditions, resources available, etc. To avoid management overload due to the control of changes, the model must adapt in a transparent and automatic way to these changes.

The previous list of characteristics poses important challenges on the underlying security mechanisms and especially in authorization and access control systems. Paradoxically, it is frequent for access control and authorization mechanisms in distributed systems to rely on centralized security administration. In fact, existing solutions for distributed authorization and access control do not provide the flexibility and manageability required. Summarizing, it is clear that new solutions are required to address the security needs of some of the new distributed applications, as it is the case of e-government, but also of web services, electronic commerce or grid computing.

The paper is organized as follows. Section 2 summarizes some background and related work. Section 3 describes the fundamentals of our proposal and outlines the system operation and implementation. Finally, section 4 summarizes the conclusions.

2 Background and Related Work

The problem of interoperation among autonomous applications has been extensively studied. For instance, it received significant attention during the late 1980s and early 1990s in the framework of the research in federated databases. The objective of this work was to resolve the structural differences among disparate database schemes. However, practical federated database systems failed because of the problem of semantic heterogeneity [1]. This problem appears when different applications mean different things by similar terms. Semantic heterogeneity is closely tied to the context-dependent interpretations of the concepts represented. Although interoperability of applications have been extensively studied (i.e., CORBA, DCom, Java, …), not much work has been done in semantic interoperation of applications. We have just to mention Web Services, providing interoperability among components with semantic heterogeneity. In this sense, a recent approach is to consider semantic aspects, applying concepts of the Semantic Web, such as ontology, to Web Services [2].

When considering the security requirements of different distributed applications, authorization often emerges as a central element in the design of the whole security system. The reason for this is that authorization is the source of the trust chain. Therefore, many security properties are determined by the flexibility, trustworthiness and expressiveness of the authorization scheme.

The problem of authorization is well known and has been studied for a long time. However, the advances in communication networks have fostered the evolution from centralized to distributed systems and applications. This situation requires the creation of new authorization models.

Currently, most authorization approaches are based on locally-issued credentials (containing attributes or privileges) that are linked to user identities. This type of credentials presents many drawbacks. Among them we highlight: (a) they are not interoperable; (b) the same credentials are issued many times for each user, what introduces management and inconsistency problems; (c) credentials are issued by the site administrator; however, in most cases, the administrator does not have enough information or resources to establish trustworthy credentials; and (d) they are tightly dependent on the user identity. But, in practice, it is frequent that the identity of the user is not relevant for the access decision. Sometimes it is even desirable that the identity is not considered or revealed. Furthermore, in systems based on identity, the lack of a global authentication infrastructure (PKI) forces the use of local authentication schemes. In these cases, subscription is required and users have to authenticate themselves to every accessed source.

Summarizing, when these local schemes are applied to distributed systems, especially to open ones, they result very limited and inconvenient. The most relevant problem when local schemes are applied to open distributed systems is the lack of interoperability. It is not reasonable to expect that heterogeneous systems for different purposes and under control of different stakeholders will be able to define a common homogeneous set of authorization criteria. Other problems are that (i) security administration is complex and error prone; (ii) allocation of policies to resources is explicit and static; (iii) access control criteria are defined either explicitly or on the basis of the location of the contents; (iv) schemes are based on user identity; and (v) access policies are dependent on the administrator of the server where the resource resides.

Based on asymmetric cryptography, digital certificates are used to bind a public key to some information. Identity certificates are the most common type of digital certificates in use today. These are used to bind identity information to keys. On the other hand attribute certificates bind attributes to keys. Therefore, attribute certificates provide means for the deployment of scalable access control systems in the scenarios that we have depicted.

The latest ITU-T X.509 recommendation [3] standardizes the concept of attribute certificate, and defines a framework that provides the basis upon which a Privilege Management Infrastructure (PMI) can be built. Precisely, the foundation of the PMI framework is the Public Key Infrastructure (PKI) framework defined by ITU [4]. This new recommendation defines a new type of authority for the assignment of privileges, the *Attribute Authority* (AA), while a special type of Authority, the *Source of Authority* (SOA), is settled as the root of delegation chains. One important point is that PKI and PMI are separate infrastructures in the sense that either structure can work on its own, or to be more precise, they can be established and managed independently.

3 Semantic Integration of PMIs in the Access Control System

The aforementioned problems related to the use of local schemes, lead us to consider a fully distributed approach. Accordingly, the inclusion of external authorization entities in the access control scheme facilitates the separation of responsibilities, enhances the security levels, and makes credentials interoperable among different access control systems.

By considering attributes to be the basis of the access control model we can develop a very flexible and open model that fits most scenarios. In fact, MAC [5], DAC [6] and RBAC [7] schemes can be specified using the attribute-based approach. In [8] we proposed a modular and dynamic approach based on the separate specification of the access control criteria and the rules of allocation of policies to resources. Additionally, the use of attributes as the central element of the model is complemented with the use of metadata to represent the semantics of the different elements in an access control system.

This new model is called Semantic Access Control (SAC) [9] because it is based on the semantic properties of the resources to be controlled, properties of the clients that request access to them, semantics about the context and finally, semantics about the attribute certificates trusted by the access control system. In SAC, access policies are expressed in terms of sets of attributes instead of users or groups. For interoperability and security reasons, client attributes must be digitally signed by a trusted certification entity external to the access control management system. Therefore, attribute certificates are used to prove that users meet the required attributes. This scheme scales well in the number of users and also in the number of different attributes used by the access control system.

In the development of the SAC model, we have considered the operation of several independent access control systems and authorization entities. In SAC, the access control to resources is independent of their location. The identification of the user or client is not mandatory. The independence of the authorization function is the key to the interoperability because it allows attributes to be safely communicated avoiding the necessity of being locally emitted by each system administrator.

Additionally, this approach avoids the registration phase of the client, and the evaluation and issuance of a client attribute repeatedly for each access control system. Finally, this scheme promotes the operation of specialized authorization entities with deep knowledge of the domain of the attribute to attest, enhancing the practical security and privacy levels of the system.

In access control schemes based on attribute certificates, the semantics of the policies depend heavily on the semantics of the attribute certificates. For this approach to be secure, a mechanism to establish the trust between these access control systems and the authorization entities is required. We have addressed this problem using semantic information about the certifications issued by each authorization entity. This mechanism is the core of the semantic integration of the PMI, which is essential in order to achieve interoperability in these scenarios. Furthermore, this integration solves the problems of separation of duties, scalability and interoperability. The main reason for this is the necessity of understanding the specific security requirements, as well as the semantics of the attribute certificates managed. As we will show, a new metadata model, called *Source Of Authorization Description* (SOAD), has been created for this purpose. The SOAD metadata model conveys the semantics of the attribute certificates providing semantic information that will be essential in the process of access decision.

The semantic information about the attribute certificates issued by each SOA also assist the security administrator through the process of specification of the access control policies, as it conveys the meaning of each attribute. Additionally, the semantic information represented by the SOAD model enables the automatic detection of inconsistent policies, through a *Semantic Policy Validator* (SPV) tool developed with this objective. The SPV makes inference processes using the rules defined in the SOAD documents.

The ability to perform a semantic validation of access control policies is an essential design goal of the SAC model. Both the *Semantic Policy Language* (SPL) defined in SAC and the semantic descriptions of the certificates issued by each SOA (conveyed by SOAD documents) are designed to serve this objective. The semantic validation ensures that the policies written by the security administrator produce the desired effects. The SPV can perform three types of validations:

1. Test Case Validation: Given a request to access a resource and a set of attribute certificates, this algorithm outputs the sets of attribute certificates needed for accessing that resource. Most of times, this feature will be used to check that a set of

attribute certificates is incompatible with the access criteria for that resource. For instance, the administrator of our university can use this validation to guarantee that it is not possible for a student to access a given resource (i.e., documents containing marks). During the validation process, the SPV generates the sets of attribute certificates that are not excluded by the input set, and checks the generated ones against all possible combinations of attribute certificates that grant access to the resource.

2. Access Validation: Given a request to access a resource, this algorithm outputs the sets of certificates that grant access to that resource. For this validation process, the SPV generates the policy for the resource and all sets of attribute certificates equivalent to those required by the policy.
3. Full Validation: The goal of this process is to check which resources can be accessed given a set of attribute certificates. Therefore, SPV generates the policy for each resource and, afterwards, all attribute certificates that can be derived from the input set of attribute certificates. Finally, it informs of every resource that can be accessed using the input attribute certificate set.

3.1 The SOAD Metadata Model

The set of SOADs represents the semantic description of the PMI. SOAD documents are digitally signed [10] XML-Schema instances expressing the different attributes certified by each SOA, including names, descriptions and relations. Such descriptions are the basis for building a mechanism to provide client applications (i.e., access control system) with knowledge about the meaning of the attributes issued by each SOA.

SOAD documents include a reference to the SOA described (`SOA_ID`), the declaration of the attribute certificates issued by that SOA and the relations between these attribute certificates.

The attribute declaration section consists in a set of `SOA_Attribute` elements. Each one of these elements defines an attribute certificate issued by the SOA referenced by `SOA_ID`, described by a name (`AttributeName`), a value (`AttributeValue`) and the signer of the certificate (`SOA_ID`).

Relations between attributes are expressed using `SOARule` elements. Each relationship is represented by a logical rule where both, the premise and the conclusion are set of attribute certificates, combined by a logical operator indicating the relation among these certificates. Each premise comprises certificates (`SOA_Attribute` elements) issued by the SOA being described or external ones. The conclusion comprises `AttributeSet` elements composed by attribute certificates issued by the SOA being described. In this way, the SOA can declare any kind of relationship among the certificates it issues and the certificates issued by other SOAs. Additionally, the client applications (i.e. access control systems) can control which relationships they accept and under which conditions.

3.2 Implementation

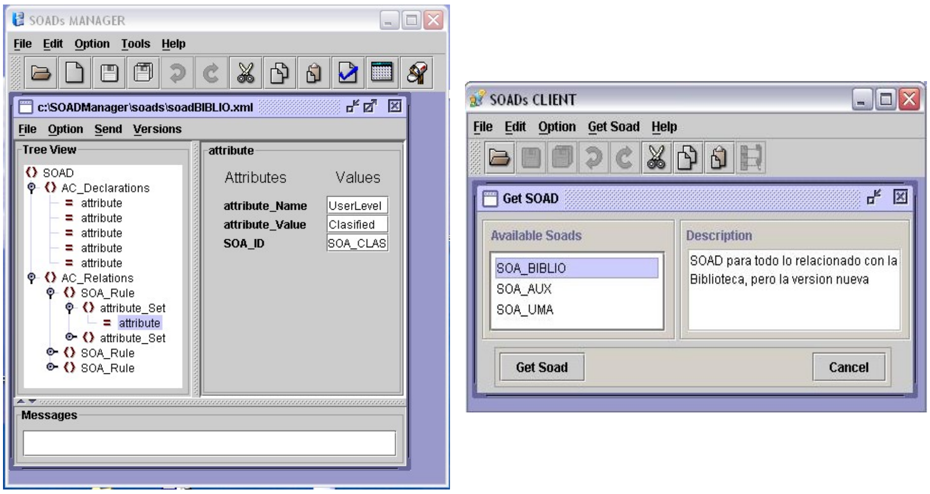
The system is implemented using three different applications: SOAD Manager, SOAD Server and SOAD Client.

The SOAD Manager is a Java™ application that allows SOAs to create SOAD documents. It has advanced edition capabilities that facilitate the definition of SOADs in an intuitive and easy way.

The SOAD Server is responsible for the publication and distribution of SOADs. This application implements an interface to allow SOAs to upload their SOADs and another one to allow clients to locate and retrieve the SOADs they need.

The principal purpose of the SOAD Client is to allow client systems to locate and retrieve SOADs from SOAD Servers. Additionally, it offers a subset of the SOAD edition capabilities available in the SOAD Manager. This application is also used to automate and tailor the process of refreshing the SOAD.

Figure 1 shows screenshots of the SOAD Manager and the SOAD Client Applications.



Attributes	Values
attribute_Name	UserLevel
attribute_Value	Clasified
SOA_ID	SOA_CLAS

Available Soads: SOA_BIBLIO, SOA_AUX, SOA_UMA

Description: SOAD para todo lo relacionado con la Biblioteca, pero la version nueva

Fig. 1. SOAD Manager and SOAD Client Applications.

3.3 System Operation

Figure 2 depicts the flow of SOADs from originating SOAs to client access control systems. Each SOA creates SOADs to describe the attribute certificates it issues. These SOADs are then made available to client systems in one or more SOAD Servers. When necessary, clients retrieve the SOADs of the SOAs they trust from SOAD Servers. Clients are then able to process the received SOADs locally in order to limit the attributes and relations they accept from each particular SOA. These local SOADs are then used in the computation of the access control decisions. Associated to each local SOAD, clients can set different parameters to control when they must be refreshed, where to refresh it from, etc.

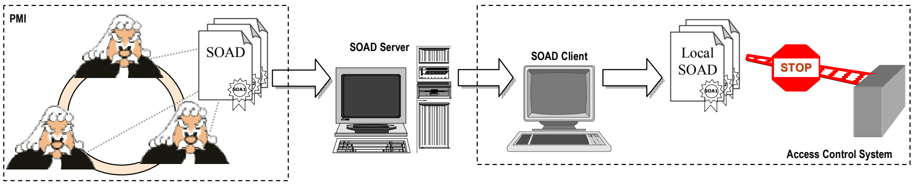


Fig. 2. Flow of SOADs

3.4 A Case Study in E-Government

We use *e-government* as the scenario to illustrate our proposal, because it is one of the most relevant and interesting of the aforementioned applications. The term *e-government* is often defined as the use of information and communication technologies to support and improve the activities of public administrations. This definition means that, to some degree, e-government is not a new issue. But the real potential of e-government lies on the possibility of substituting traditional paper-based procedures by their electronic versions, achieving what have been called "paperless systems", implementing the necessary mechanisms to achieve trustful and transparent interoperation between the different parties involved (government agencies, citizens, private businesses and organizations, other arms of government and even foreign governments).

Figure 3 shows a typical e-government scenario. In particular, we use a representative example, involving the interaction of several government agencies, some private business and organizations as well as individual citizens. Consider the case of a tax collection agency starting a judicial process against a citizen, due to unpaid taxes. This process implies different exchanges of sensitive information among different parties. The judge from the corresponding court can request information to the Town Hall cadastral agency about the cadastral value of the buildings belonging to the citizen accused, to the bank about the drawing account of the citizen, to the police department about the criminal records of the accused citizen, etc. On the other hand, the accused citizen and his lawyer may request information about the stage of the judicial process.

Interoperability is an essential requirement in this scenario. The existence of different government agencies that need to cooperate, and the special security requirements inherent to these transactions, makes this problem very complex. In order to securely perform this information exchange, each party has to be recognized as an official entity with jurisdiction to do the intended task. Identity-based schemes are not always the best option. Every single piece of information in the different sites has different requirements making it impossible in practice to assign privileges to identities. In this case, the authorization of the other party (i.e., the examining judge) is based on some specific properties or attributes (to be the judge assigned to the process), not on identity (to be Mr. Jones). These properties represent the conditions that the user (or the client agency requesting the service) has to fulfil in order to access the information, that is, the access control policy.

In this scenario, an access control model based on attributes is very appropriate and can provide simple solutions to such problems. But, the real advantage comes when attributes become interoperable. To achieve such interoperability, we must satisfy two important conditions. First, attributes must be come from trusted sources, and second, we must be able to understand what those attributes mean.

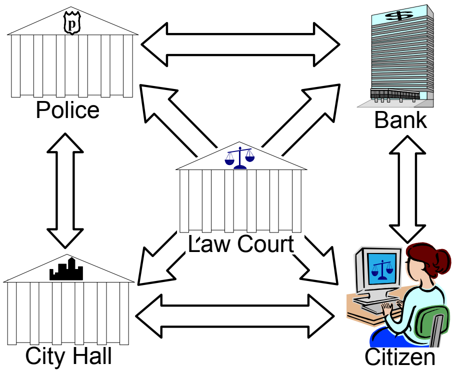


Fig. 3. An e-government scenario

Our proposal fulfils the requirements of this kind of transactions, providing fine-grained access control, enabling the secure communication among government agencies and assigning the attestation of attributes to trusted entities with an in-depth knowledge of the properties to attest (SOA of the Policy Department, SOA of the Law Court, etc.).

4 Conclusions

The possibility of automating the processing of semantic information is a big challenge for the resolution of many relevant problems, as is the case of semantic interoperability. The objective of this work is to reach semantic interoperability through semantic integration in distributed environments, where remote and heterogeneous parties must exchange information in a controlled manner. We think the development of mechanisms for the semantic integration in distributed environments where heterogeneity is common, implies the development of semantic models supported by metadata infrastructures. In the case we are concerned with, the kind of information to be described is essential to maintain the secure, trustful and transparent interoperation of the different parties involved in electronic transactions.

We have presented a solution for the interoperability of authorizations (attribute certificates) based on the description of their semantics. This solution provides a foundation to build interoperable access control systems with external and independent authorization services. Additionally, the semantic modelling of the authorizations enables interesting possibilities, such as the semantic validation of access control policies.

References

1. Sheth, A., Larson, J.: Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases. ACM Computing Surveys, 22(3) (1990) 183 - 236
2. World Wide Web Consortium: Semantic Web Services Interest Group. Retrieved January 2003 from http://www.w3.org/2002/ws/swsig/
3. International Telecommunication Union (2000). ITU-T Recommendation X.509. Information technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks. Technical Cor. 3 (02/03) [Electronic version] http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-X.509-200302-P!Cor3
4. International Telecommunication Union (1997). ITU-T Recommendation X.509, Information Technology – Open systems interconnection – The Directory: Authentication Framework. 1997. [Electronic version] http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-X.509-200302-T!Cor5
5. Qian, X., Lunt, T.F.: A MAC policy framework for multilevel relational databases. IEEE Transactions on Knowledge and Data Engineering, 8(1) (1996) 1-14
6. Baraani, A., Pieprzyk, J., Safavi-Naini, R.: Security In Databases: A Survey Study. Retrieved September 2003 from [http://citeseer.nj.nec.com/baraani-dastjerdi96security.html. (1996)
7. Sandhu, R., Ferraiolo, D., Kuhn, R.: The Nist model for role-based access control: Towards a unified standard. In Proceedings of 5th ACM Workshop on Role-Based Access Control. Berlin, Germany (2000)
8. López, J., Maña, A. and Yagüe, M.I: XML-based Distributed Access Control System. Lecture Notes in Computer Science, Vol. 2455. Springer-Verlag (2002)
9. Yagüe, M.I., Maña, A., López, J., Pimentel, E., Troya, J.M.: A Secure Solution for Commercial Digital Libraries. Online Information Review Journal, 27(3): 147-159. Emerald Publishers (2003)
10. World Wide Web Consortium: XML-Signature Syntax and Processing (2002) [Electronic version] http://www.w3.org/TR/xmldsig-core/.

Promiscuous Mode Detection Platform

Zouheir Trabelsi and Hamza Rahmani

College of Telecommunications
The University of Tunisia
Cité Technologique des Communications
Route de Raoued Km 3,5 – 2083 El Ghazala, Ariana, Tunisia
trabelsi.zouheir@supcom.rnu.tn

Abstract. Among various types of attacks on an Ethernet network, "sniffing attack" is probably one of the most difficult attacks to handle. Sniffers are programs that allow a host to capture any packets in an Ethernet network, by putting the host's Network Interface Card (NIC) into the promiscuous mode. When a host's NIC is in the normal mode, it captures only the packets sent to the host. Since many basic services, such as FTP, Telnet and SMTP, send passwords and data in clear text in the packets, sniffers can be used by hackers to capture passwords and confidential data.

A number of anti-sniffers have been developed, such as PMD [18], PromiScan [17] and L0pht AntiSniff [19]. An anti-sniffer is a program that tries to detect the hosts running sniffers, in a Local Area Network (LAN). Current anti-sniffers are mainly based on three detection techniques, namely: the ARP detection, the DNS detection, and the RTT (Round Trip Time) detection techniques [13 and 16]. However, sniffers are becoming very advanced so that anti-sniffers are unable to detect them. The main drawback of these detection techniques is that they rely on the ARP, ICMP and/or DNS reply messages generated by the sniffing hosts. Therefore, in order to stay undetectable by anti-sniffers, advanced sniffers do not generate such reply messages while sniffing.

This paper discusses an anti-sniffer based on a new detection technique. The technique uses mainly ARP cache poisoning attack to detect sniffing hosts in an Ethernet network. The technique is implemented in a tool, called SupCom anti-sniffer, which automatically gives system administrator a better helping hand regarding the detection of sniffers. Four anti-sniffers, PMD [18], PromiScan [17], L0pht AntiSniff [19] and SupCom anti-sniffer, are tested and the evaluation results show that SupCom anti-sniffer succeeded to detect more sniffing hosts than the other anti-sniffers.

1 Introduction

Malicious users can easily steal confidential documents and anyone's privacy by sniffing a network. It can be done simply by downloading free sniffer software from the Internet and installing it into a personal computer (PC). Sniffers capture all packets in a network. To achieve this, the sniffer sets the Network Interface Card (NIC) of the computer into a mode called "promiscuous mode". Then the NIC will blindly receive all packets and pass them to the system kernel. Packets that are not supposed to arrive to that PC are no longer blocked by the NIC. Those PC's with promiscuous NIC's are running sniffers.

Many basics services, such as FTP, Telnet and SMTP [9], send clear text data in the packets. A sniffer captures all packets and displays their contents on the hacker's computer screen, for examples the passwords used to authenticate during an FTP session, or the message of an email in SMTP packets. Hackers can spy the users of a network, just by reading and analyzing the contents of the packets going to and out of the users' hosts. This type of attack on a network is usually difficult to detect, since it does not interfere with the network traffic at all. System administrators are facing difficulties to detect and deal with this attack.

In this paper, we first explore three different techniques used to detect sniffing hosts in an Ethernet network, and we discuss their limits. The three techniques are: the ARP detection technique, the DNS detection technique and the RTT detection technique. Most anti-sniffers, such as PMD [18], PromiScan [17] and L0pht AntiSniff [19], are based on these detection techniques. However, the techniques present many drawbacks so that advanced sniffers are designed in such a way they can stay undetectable by anti-sniffers.

Then, we discuss an anti-sniffer based on a new detection technique. The technique includes mainly three phases and uses ARP cache poisoning attack to detect sniffing hosts, in an Ethernet network. Based on this technique, a tool, called SupCom anti-sniffer, is implemented. SupCom anti-sniffer gives automatically system administrators a better helping hand regarding the detection of sniffers. Four anti-sniffers, PMD[18], PromiScan[17], L0pht AntiSniff [19] and SupCom anti-sniffer, are tested and the evaluation results show that SupCom anti-sniffer gives a better detection performance than the others.

2 Basic knowledge

2.1 NIC's hardware addresses

All the NIC cards on the Ethernet are represented by a 6-bytes hardware address (MAC address). The manufacturer assigns this address such that each address is unique in the whole world. Theoretically, there are no two NIC's having the same hardware address. All communications on the Ethernet are based on this hardware address. The NIC, however, can set up different filters (called hardware filter) in order to receive different kinds of packets. The following are a list of hardware filters:

- *Broadcast*: Receive all broadcast packets. Broadcast packets have destination address FF:FF:FF:FF:FF:FF. The purpose of this mode is to receive the packets which are supposed to arrive at all nodes existing on the network.
- *Multicast*: Receive all packets which are specifically configured to arrive at some multicast group addresses. Only packets from the hardware multicast addresses registered beforehand in the multicast list can be received by the NIC.

- *All Multicast*: Receive all multicast packets. Since this mode may also correspond to other high level protocols other than IPv4, all Multicast will receive all packets that have their group bit set (01:00:00:00:00:00).
- *Promiscuous*: Receive all packets on the network without checking the destination address at all.

2.2 ARP messages

ARP messages are exchanged when one host knows the IP address of a remote host and wants to discover the remote host's MAC address. For example, if host1 wants host2's MAC address, it sends an ARP request message (Who has?) to the broadcast MAC address (FF:FF:FF:FF:FF:FF) and host2 answers with his addresses (MAC and IP). Basically, an ARP message on an Ethernet/IP network has 8 important parameters:

- Ethernet header:
    - Source MAC address
    - Destination MAC address
    - Ethernet Type (=0x0806 for ARP message)
- ARP message header:
    - Source IP address
    - Source MAC address
    - Destination IP address
    - Destination MAC address
    - Operation code:
        - 1: for ARP request
        - 2: for ARP reply.

It is important to mention that there is nothing specifying that there must be some consistency between the ARP header and the Ethernet header. That means you can provide uncorrelated addresses between these two headers. For example, the source MAC address in the Ethernet header can be different from the source MAC address in the ARP message header.

3 Related Work

3.1 The RTT detection technique

The RTT (Round Trip Time) is the time of the round trip of a packet sent to a host. That is the time that a packet took to reach the destination, plus the time that a response took to reach the source. It is expected that the measurement of the RTT increases considerably when a host is in the promiscuous mode, since all packets are captured.

The idea behind the RTT detection technique ([16] and [13]), is first to send to a host, with a particular OS, a number of request packets, and wait for the responses packets, in order to take the RTT measurements. Then, the host is set to the promiscuous mode. And, the same request packets are sent again to the host, and the corresponding RTT measurements are collected. The RTT averages, the standard deviations, and the percentage of changes of the collected RTT measurements are computed. The RTT averages, standard deviations, percentage of changes are called the training data.

The samples of the collected RTT measurements represent two different populations, called the normal mode population and the promiscuous mode population. To show that the two averages of the samples RTT measurements are statistically different enough and therefore represent two different populations (the normal mode and the promiscuous mode populations), the z-statistics [1] model is used. The z-statistics model allows to make a judgment about whether or not a host's NIC is set to the promiscuous mode.

In the real world, the system administration has to identify first the OS of the suspicious host. This can be done by several available tools, such as Nmap [15]. Then, a number of request packets should be sent to the suspicious host in order to collect the corresponding RTT measurements.

The suspicious host can be either in the normal mode or in the promiscuous mode. Two z-statistics are computed. The first one, called the normal mode z-statistics, uses the training data related to the OS of the suspicious host for the normal mode, as the first population, and the collected data in the real world, as the second population. The second z-statistics, called the promiscuous mode z-statistics, uses the training data related to the OS of the suspicious host for the promiscuous mode, as the first population, and the collected data, as the second population. If the normal mode z-statistics is less than the z value (which is 2.36), then we may conclude that the host's NIC is almost 99% set to the normal mode, else, the host's NIC is set to the promiscuous mode.

***The limits of the RTT detection technique*:** The RTT detection technique is a probabilistic technique. Many known and unknown factors, such as the operating system of the suspicious host, and the LAN traffic, may affect considerably the results generated by any anti-sniffer based on this technique. When the LAN is under heavy traffic, this probabilistic technique may generate false decision regarding whether the suspicious host's NIC card is set to the promiscuous mode or to the normal mode. This is due mainly on the RTT measurements taken which may lead to a false decision. In addition, an advanced sniffer may attempt to put heavy traffic in the network in order to let the anti-sniffer generates misleading results.

The RTT detection technique attempts to send heavy traffic to a suspicious host on a particular open port, usually the FTP port (21). However, it is not common to have always the FTP port (21) open in each host in the network. Finally, to work appropriately, this technique needs to send heavy traffic on the network and then takes the RTT measurements. Such an action may cause some damage to the network's hosts and services, such as denial of service attacks.

3.2 The DNS detection technique

The DNS detection technique [13] works by exploiting a behavior common to many sniffers. Current sniffers are not truly passive. In fact, current sniffers do generate network traffic, although it is usually hard to distinguish whether the generated network was from the sniffer or not. It turns out that many sniffers do reverse DNS lookup (that is looking up a hostname by an IP address) on the traffic that it sniffed. Since this traffic is generated by the sniffer program, the trick is to detect this DNS lookup some how and distinguish it from normal DNS lookup requests.

To do that, we can generate fake traffic to the Ethernet segment with a source address of some unused IP address. Then, since the traffic we generate should normally be ignored by the hosts on the segment, if a DNS lookup request is generated, we know that there is a sniffer on the Ethernet segment. And by sniffing the packets on the Ethernet segment, we can detect which hosts are sending the DNS lookup requests.

The limits of the DNS detection technique: This technique can be quickly side stepped. Sniffers can easily be changed to not perform the reverse DNS lookup. Furthermore, hackers will become more intelligent so as to never perform the reverse DNS lookup either. This will render the technique completely useless.

3.3 The ARP detection technique

The ARP detection technique is described more in detail in our paper [16]. However, we need to describe it again here since; this paper uses some of its results.

The ARP detection technique consist into checking whether or not a suspicious host responds to ARP request packets that are not supposed to be treated by the suspicious host. Since the sniffing host receives all the packets, including those that are not targeting to it, it may make mistakes such as responding to a packet, which originally is supposed to be filtered by the host's NIC. Therefore, the detection is performed by checking the responses of ARP reply packets, when ARP request packets are sent to all hosts on the network.

On an Ethernet linked by IP addresses, packets are in fact sent and received based on hardware addresses (MAC address). Packets cannot be sent by just using an IP address. Therefore, the Ethernet needs a mechanism that converts IP addresses into hardware addresses. At this time, ARP packets are used. ARP packets belong to the link layer, which is the same layer as IP, so ARP packets does not affect the IP layer. Since IP addresses resolving is always available on an IP network, ARP packets become suitable packets for testing the response of the hosts when detecting promiscuous mode.

3.3.1 Promiscuous mode detection

When the NIC is set to promiscuous mode, packets that are supposed to be filtered by the NIC are now passed to the system kernel. Therefore, if we configure an ARP packet such that it does not have broadcast address as the destination address, send it to every host on the network and discover that some hosts respond to it, then those hosts are in promiscuous mode.

In this example, the ARP packet destination hardware address is set to an address that does not exist, for example 00-00-00-00-00-01. When the NIC is in normal mode, this packet is considered to be “to other host” packet, so it is refused by the hardware filter of the NIC. However, when the NIC is in promiscuous mode, the NIC does not perform any filter operation. Then this packet is able to pass to the system kernel. The system kernel assumes that this ARP requests packet arrives because it contains the same IP address as that machine, so it should respond to the packet. However, this is not true. There exists some sort of software filter in the kernel, called the Software Filter, because a packet is actually filtered again by the system kernel. The software filter depends on the operating system kernel.

3.3.2 Software filtering based detection

It is unnecessary to sent ARP packet with MAC addresses that do not exist, since the software filter will block such packets. However, we need to send ARP packets with MAC addresses that may pass the software filter. So that, we can understand the mechanism used by the software filter to filter packets based on their MAC addresses. The following are the list of hardware MAC addresses used to send ARP request packets, when the NIC is in the promiscuous mode (the hardware filter do not filter packets):

- FF:FF:FF:FF:FF:FF broadcast address :
  All nodes should receive this kind of packets and respond because it is a broadcast address. A usual ARP request packet uses this address.
- FF:FF:FF:FF:FF:FE fake broadcast address : This address is a fake broadcast address missing the last 1 bit. This is to check whether the software filter examines all bits of the address and whether it will respond.
- FF:FF:00:00:00:00 fake broadcast 16 bits : This address is a fake broadcast address in which only the first 16 bits are the same as the broadcast address. This may be classified as a broadcast address and replied when the filter function only checks the first word of the broadcast address.
- FF:00:00:00:00:00 fake broadcast 8 bits : This address is a fake broadcast address in which only the first 8 bits are the same as the broadcast address. This may be classified as a broadcast address and replied when the filter function only checks the first byte of the broadcast address.
- F0:00:00:00:00:00 fake broadcast 4 bits : This address is a fake broadcast address in which only the first 4 bits are the same as the broadcast address. This may be classified as a broadcast address and replied when the filter function only checks the first 4 bits of the broadcast address.
- 01:00:00:00:00:00 group bit address : This is an address with only the group bit set. This is to check whether this address is considered as a multicast address as Linux does.
- 01:00:5E:00:00:00 multicast address 0 : Multicast address 0 is usually not used. So we use this as an example of a multicast address not registered in the multicast list of the NIC. The hardware filter should reject this packet. However, this packet may be misclassified to be a multicast address when the software filter does not completely check all bits. The system kernel thus may reply to such packet when the NIC is set to promiscuous mode.

- 01:00:5E:00:00:01 multicast address 1 : Multicast address 1 is an address that all hosts in the local network should receive. In the other word, the hardware filter will pass this kind of packets by default. But it is possible that the NIC does not support multicast mode and does not respond, but this hypothesis was not available because all the available cards on the market bear multicasting. So this is to check whether the host supports multicast addresses.
- 01:00:5E:00:00:02 multicast address 2 : Multicast address 2 is used to all routers in the local networks. So we use this as an example of a multicast address not registered in the multicast list of the NIC. The hardware filter should reject this packet and also is not accepted by the software filter. The system kernel check the hardware result and one notices while the software filter always comes after the hardware filter, from which for the addresses multicast, if an address was rejected by the hardware filter she is therefore rejected by the software filter.

01:00:5E:00:00:03 multicast address 3 : Multicast address 3 is not assigned. So we use this as an example of a multicast address not registered in the multicast list of the NIC. The hardware filter should reject this packet and also is not accepted by the software filter. The system kernel check the hardware result and one notices while the software filter always comes after the hardware filter, from which for the addresses multicast, if an address was rejected by the hardware filter she is therefore rejected by the software filter.

3.3.3 Experiences and results

The tests are performed against a number of operating systems (Windows 9x, ME, 2000/NT and XP, Linux 2.4x and FreeBSD 5.0). As expected, all kernels respond to the broadcast address and multicast address 1 when the NIC is in normal mode. The test results using the hardware addresses listed in the previous section, are listed in Table 1.

However, when the NIC is set to the promiscuous mode, the results are OS dependent.

Microsoft Windows :

- In the case of Windows 9x and ME, it responds to fake broadcast addresses B47, B16, and B8. Hence, the software filters of Windows 9x and Me determine the broadcast address by checking only the first byte. Because when we test with fake address F0:00:00:00:00:00, it will not response, so the mechanism of check, try to check only FF:??:??:??:??:??. Therefore, the three addresses B47, B16 and B8 can be used to verify whether a NIC card is set to a promiscuous mode or not. If the NIC is in the promiscuous mode, it will responds to an ARP request packet, by an ARP reply packet.
- In the case of Windows 2000/NT, it responds to fake broadcast B47 and B16. Hence, the software filters of Windows 2000/NT determine the broadcast address by checking only the 2 first bytes. Since Windows 2000/NT responds to the fake broadcast B16 in the normal mode also, therefore, only the addresses B47 can be used to verify whether a NIC card is set to a promiscuous mode or not.

- In the case of Windows XP, it responds to fake broadcast addresses B47 and B16. Hence, the software filter of Windows XP determines the broadcast address by checking only the first two byte. Therefore, the two fake broadcast addresses B47 and B16 can be used to verify whether a NIC card is set to a promiscuous mode or not.

Linux and FreeBSD :

In the case of Linux 2.4x and FreeBSD 5.0, it responds to all fake broadcast and to all addresses with the group bit set. Therefore, any fake broadcast addresses can be used to very the promiscuous mode. In addition, any address with the group bit set can be used to verify the promiscuous mode, excluding the multicast address M1. Since, Multicast address M1 is an address that all hosts in the local network should receive.

Operating Systems / Hardware Addresses		Windows XP		Windows Me/9x		Windows 2k/NT		Linux 2.4.x		FreeBSD 5.0	
		Norm.	Prom.	Norm.	Prom.	Norm.	Prom.	Norm.	Prom.	Norm.	Prom.
FF:FF:FF:FF:FF:FF	Br	O	O	O	O	O	O	O	O	O	O
FF:FF:FF:FF:FF:FE	B47	--	X	--	X	--	X	--	X	--	X
FF:FF:00:00:00:00	B16	--	X	--	X	**X**	**X**	--	X	--	X
FF:00:00:00:00:00	B8	--	--	--	X	--	--	--	X	--	X
01:00:00:00:00:00	Gr	--	--	--	--	--	--	--	X	--	X
01:00:5E:00:00:00	M0	--	--	--	--	--	--	--	X	--	X
01:00:5E:00:00:01	M1	O	O	O	O	O	O	O	O	O	O
01:00:5E:00:00:02	M2	--	--	--	--	--	--	--	X	--	X
01:00:5E:00:00:03	M3	--	--	--	--	--	--	--	X	--	X

O**: Legal response,** X**: Illegal response, --: No response**

3.3.4 The limits of the ARP detection technique

The main limits of this detection technique is that if a host does not generate any ARP reply messages while sniffing, then this technique becomes useless. Because this detection technique relies on the ARP reply messages generated by the sniffing host. Consequently, any anti-sniffer based on this detection technique is unable to detect the sniffing hosts that do not generate ARP reply messages.

4 ARP cache poisoning attack based detection technique

4.1 ARP cache

Each host in a network segment has a table, called ARP cache, which maps IP addresses with their MAC addresses. New entries in the ARP cache can be created or already existing entries can be updated by ARP request or reply messages.

Create a new entry: When an ARP reply message arrives in a host, an entry in the ARP cache should be created. If the entry exists already, then it should be updated. In addition, when a host receives an ARP request message, it believes that a connexion is going to be performed. Hence, to minimize the ARP traffic, it creates a new entry in its ARP cache and puts there the addresses provided in the ARP request message. It is important to mention that sending an ARP request message in unicast is totally RFC compliant. They are authorized to let a system checks the entries of its ARP cache.

Update an entry: When an ARP reply message or an ARP request message arrives in a host, if the entry exists already, then it will be updated by the addresses (the source MAC address and the source IP address) provided in the ARP message.

4.2 ARP cache poisoning attack

ARP cache poisoning attack is the malicious act, by a host in a LAN, of introducing a spurious IP address to MAC address mapping in another host's ARP cache. This can be done by manipulating directly the ARP cache of a target host, independently of the ARP messages sent by the target host. To do that, we can either:

- ➢ add a new fake entry in the target host's ARP cache
- ➢ or, update an already existing entry by fake addresses (IP and/or MAC addresses).

Create a fake new entry: To do that, we send an ARP request message to a target host, with fake source IP and MAC addresses. When the target host receives the ARP request message, it believes that a connexion is going to be performed, and then, creates a new entry in its ARP cache and puts there the fake source addresses (IP and/or MAC) provided in the ARP request message. Consequently, the target host's ARP cache becomes corrupted.

Update an entry with a fake entry: To do that, we just have to send an ARP reply message to a target host with fake IP and MAC addresses. Thus, even if the entry is already present in the target host's ARP cache, it will be updated, with the fake entries.

4.3 Sniffing hosts' detection technique

The proposed detection technique used to detect sniffing hosts is based mainly on the ARP cache poisoning attack. It consists of three different phases:

- In the first phase, we attempt to corrupt the ARP cache of each sniffing host in the LAN, with a fake entry, using ARP cache poisoning attack. We will demonstrate that only the ARP caches of the hosts running sniffers will be corrupted, and this attack on the ARP caches will not make any damage to the attacked hosts.

- In the second phase, we attempt to establish a TCP connexion with each host in the LAN on any port, whether it is an open port or a closed one.
- In the third phase, we sniff the LAN in order to capture any packet containing the fake entry. We will demonstrate that the hosts that sent TCP or ICMP packets containing the fake entry are running sniffers. However, the hosts that sent ARP request packets are not running sniffers.

The following sub-sections describe in detail the three phases. We assume that we use a host in the LAN, called the testing host, to do all the actions needed in the three phases.

4.3.1 Phase 1: ARP cache poisoning

The aim of this phase is to corrupt only the ARP caches of the sniffing hosts in a LAN. First, we send an ARP request message, with fake source IP and MAC addresses (IP-X and MAC-X), to all hosts in the LAN.

The ARP request message sent to the hosts has an Ethernet header and an ARP message header. Hence, we need to choose the values of the fields in each header, in order to let only the sniffing host processes the ARP request message. If we choose the destination MAC address in the Ethernet layer header as a broadcast address (*FF:FF:FF:FF:FF:FF)*, then all the ARP caches of the hosts in the LAN will be corrupted by the ARP cache poisoning attack. Such a destination MAC address is discarded because it does not allow us to detect which hosts are sniffing.

However, if the destination MAC address is the fake broadcast address B47 (*FF:FF:FF:FF:FF:**FE***), then any host, with any OS, set to the promiscuous mode will accept the ARP request message and send it to the ARP layer (refer to section 3.3.3). If the host is set to the normal mode, this ARP request message will be blocked at the Ethernet layer, since the destination MAC address is not an unicast address, a broadcast address nor a multicast address. Consequently, the values of the main fields of the ARP request packet used to corrupt only the ARP caches of the sniffing hosts are:

- **Ethernet header:**
  - Source MAC address = *Any MAC address*
  - Destination MAC address =
    *FF:FF:FF:FF:FF:**FE** (B47)*
  - Ethernet Type = *0x0806 (ARP message)*

- **ARP message header:**
  - Source IP address = *Fake IP address (IP-X)*
  - Source MAC address =
    Fake MAC address (MAC-X)
  - Destination IP address = *IP address of a target host in the LAN*
  - Destination MAC address = *00:00:00:00:00:00*
  - Operation code: *1 (ARP request)*

4.3.2 Phase 2: Establishing TCP connections

Then, for each host in the LAN, we will attempt to establish a TCP connection. To do that, we need to send TCP packets with the bit SYN set, from the testing host, to each host in the LAN. However, the source IP address in the IP header of the TCP packets is not the source IP address of the testing host. But, it is that fake IP address (IP-X). Each host in the LAN will process the received TCP packet.

The values of the some important fields of the TCP packet used to establish a TCP connexion with each host in the LAN are:

- **Ethernet header:**
  - Source MAC address =
    The Testing host's MAC address
  - Destination MAC address =
    Target host's MAC address

- **IP header:**
  - Source IP address = *Fake IP address (IP-X)*
  - Destination IP address =
    IP address of a target host

- **TCP header:**
  - *Destination Port = Any number between 1 and 65535 (for example: 40000)*
  - *Source Port = Any number*
  - *Bit SYN = 1*

4.3.3 Phase 3: Detection of the sniffing hosts

Just following the request for establishing a TCP connexion with each host in the LAN, we expect three types of possible replies packets come from the hosts.

- The first type can be a TCP packet indicating that the connexion can be done (the SYN and ACK bits are set).
- The second type can be an ICMP error message indicating that the connexion cannot be established because the port destination is inaccessible.
- The third type can be an ARP request message sent by a host to look for the MAC address of the fake source IP address IP-X.

The hosts that generate any TCP or ICMP reply packets with the fake addresses IP-X and MAC-X as the destination addresses in the IP header are consequently running sniffers. Because, those host's ARP caches are corrupted with the fake IP and MAC addresses (IP-X and MAC-X) and are able to provided the MAC address MAC-X of the IP address IP-X. It is important to indicate again that during the first phase we used the ARP cache poisoning attack to corrupt only the ARP caches of the sniffing hosts, with the fake entry (IP-X and MAC-X).

We use a sniffer to capture any TCP or ICMP packet on the LAN that has those fake IP and MAC addresses (IP-X and MAC-X) as the destination addresses, and has been sent by a host. All hosts that sent such TCP or ICMP packets are consequently running sniffers, and their IP addresses can be easily identified.

However, any host whose ARP cache is not corrupted would generate an ARP request message in order to get the MAC address of the fake IP address IP-X. This MAC address will be used later to send the reply message which is expected to be a TCP or ICMP packet. Therefore, any host in the LAN that will send ARP request message looking for the MAC address of the IP address IP-X are not running sniffers.

4.4 Discussion

The proposed detection technique does not rely on the DNS, ARP and ICMP messages generated by the sniffing hosts. In a LAN, even an advanced sniffer cannot stay undetectable by an anti-sniffer based on the proposed ARP cache poisoning attack detection technique. Unless the sniffer stops all types of traffic directed to and issued from the sniffing host. In such a situation, the sniffer becomes useless, since no other networking activities can be done while the sniffer is working.

4.5 Implementation

Based on the proposed ARP cache poisoning attack detection technique, an anti-sniffer with a Graphical User Interface (GUI), called SupCom anti-sniffer, has been developed using Visual C++6.0 and WinpCap Library. The anti-sniffer integrates a TCP and ARP packet generator and a sniffer with filtering capabilities. SupCom anti-sniffer allows to generate ARP request packet with fake source IP and MAC addresses. In addition, it is able to sniff the network and capture packets based on filtering rules defined by the users.

SupCom anti-sniffer uses the three phases discussed in the previous sections, to detect the sniffing hosts in a LAN. SupCom anti-sniffer is able to detect hosts running even advanced sniffers. Advanced sniffers are sniffers that do not send any ARP request and reply messages, ICMP and DNS messages, in order to stay undetected by current anti-sniffers.

4.6 Evaluation

Four anti-sniffers, PromiScan [17], PMD [18], L0pht AntiSniff [19], and SupCom anti-sniffer are used to detect sniffing hosts in a LAN, during two tests. In the first test, the sniffing hosts can generate ARP reply messages. The following table, Table 2, shows that all the four anti-sniffers are able to detect all the sniffing hosts. In the second test, the hosts are running an advanced sniffer that does not generate any ARP messages and reverse DNS lookup messages. The following table shows that only SupCom anti-sniffer was able to detect all the sniffing hosts. This experience demonstrates clearly that SupCom anti-sniffer is more efficient than current anti-sniffers, particularly when detecting advanced sniffers.

Table 2: Detection performance of some anti-sniffers

Anti-Sniffers	Test 1: simple sniffer (1)	Test 2: advanced sniffer (2)
PromiScan	All sniffing hosts detected	No sniffing hosts detected
PMD	All sniffing hosts detected	No sniffing hosts detected
L0pht AntiSniff	All sniffing hosts detected	No sniffing hosts detected
SupCom anti-sniffer	All sniffing hosts detected	**All sniffing hosts detected**

(1) ***Simple sniffer***: is a sniffer that allows the sniffing host to generate all type of ARP, ICMP, TCP, UDP, and reverse DNS lookup packets.

(2) ***Advanced sniffer***: is a sniffer that does not allow the sniffing host to generate ARP packets and reverse DNS lookup packets, in order to avoid detection by current anti-sniffers.

5 Conclusion

Today, the need for techniques and anti-sniffers to detect sniffing hosts in a network is unquestionable. Hackers do not need advanced knowledge about TCP/IP protocols or networking to sniff a network. Hackers are just downloading sniffers from the Internet, and using them to spy their target hosts, and steal confidential information.

Current anti-sniffers use many detection techniques, mainly the RTT detection technique, the DNS detection technique, and the ARP detection technique. These techniques have many drawbacks, so that well designed and implemented sniffers can stay undetectable by current anti-sniffers. When the sniffing hosts do not generate any reply ARP and DNS messages, or put heavy traffic on the network, these detection techniques become useless.

In this paper, we discussed a new detection technique based mainly on ARP cache poisoning attack. We demonstrated that an anti-sniffer based on this detection technique is more effective than current anti-sniffers. The experience shows that when the sniffers do not generate any ARP reply and DNS messages, or put continuously heavy traffic on the network, only an anti-sniffer based on the proposed detection technique can detect such sniffers.

Even though sniffers are difficult to detect, the technique can provide system administrator with a consistent decision. However, by combining many detection techniques in a single anti-sniffer, systems administrators will have more results that confirm whether or not a target host is running a sniffer.

References

1. Freedman, Pisani, Purves and Adhikari, “Statistics – Second Edition”, W.W. Norton & Company, Inc. 1991.
2. Grundshober, S. “Sniffer Detector Report”, Global Security Analysis Lab., Zurich Research Laboratory, IBM Research Division, June 1998.
3. Hornig, C., “A Standard for the Transmission of IP Datagrams over Ethernet Networks”, RFC-894, Symbolics Cambridge Research Center, April 1984.
4. Plummer, David C., “An Ethernet Address Resolution Protocol-Converting Network Protocol to 48 bit Ethernet Address for Transmission on Ethernet Hardware”, RFC-826, November 1982.
5. Postel, J., “Internet Protocol”, RFC-791, USC/Information Science Institute, 1981.
6. Postel, J., ”Transmission Control Protocol”, RFC-793, USC/Information Science Institute, 1981.
7. Postel, J., ”Internet Control Message Protocol”, RFC-792, USC/Information Science Institute, 1981.

9. Richard Stevens – "TCP/IP Illustrated : Volume 1", 2001.
10. Security Software Inc., "Antisniff", Technical Report 2000, "http://www.securitysoftwaretech.com",
11. S. Grundschober. "Sniffer Detector Report", Diploma Thesis, IBM Research Division, Zurich Research Laboratory, Global Security Analysis Lab, June 1998.
12. J. Drury., "Sniffers: What are they and how to protect from them", November 11, 2000. http://www.sans.org/.
13. D. Wu and F. Wong., "Remote Sniffer Detection". Computer Science Division, University of California, Berkeley. December 14, 1998.
14. Daiji Sanai, "Detection of Promiscuous Nodes Using ARP Packets", http://www.securityfriday.com/.
15. Nmap Tools, http://securityfocus.com.
16. Zouheir Trabelsi, and all, "Malicious Sniffing Systems Detection Platform", The IEEE/IPSJ 2004 International Symposium on Applications and the Internet (SAINT2004)", Tokyo, Japan, January 26-30, 2004.
17. PromiScan anti-sniffer: "http://www.securityfriday.com".
18. PMD (Promiscuous Mode Detector): "http://webteca.port5.com".
19.L0phtAntiSniff:"http://www.l0pht.com/antisniff/"

Web Services Security: is the problem solved?

Carlos Gutiérrez[1], Eduardo Fernández-Medina[2] and Mario Piattini[2]

(1) Sistemas Técnicos de Loterías del Estado.
Calle Manuel Tovar 9, 28034, Madrid. (SPAIN). Tel: 34 91 348 92 61
carlos.gutierrez@stl.es
(2) Alarcos Research Group. Universidad de Castilla-La Mancha.
Paseo de la Universidad 4, 13071, Ciudad Real. (SPAIN). Tel: 34 926 29 53 00
{Eduardo.FdezMedina, Mario.Piattini}@uclm.es

Abstract. During the past years significant standardization work in web services technology has been made. As a consequence of these initial efforts, web services foundational stable specifications have already been delivered. Now, it is time for the industry to standardize and address the security issues that have risen from this paradigm. Great activity is being carried out on this subject. This article demonstrates, however, that a lot of work needs to be done in web services security standardization. It explains the new web services security threats and mentions the main initiatives and their respective specifications that try to solve them. Unaddressed security issues for each specification are stated. In addition, current general security concerns are detailed and a general solution is proposed.

1 Introduction

Recently web services technology has reached such a level of maturity that it has evolved from being a promising technology to becoming a reality on which IT departments are basing their operations to achieve a direct alignment with the business operations that they support [9]. In fact, based on the most recent reports from IDC[17], approximately 3300 web services-based technology projects were deployed all over North America in 2002 and it is expected that the expenses will approach $3 billion in 2003. This seems to be a logical consequence of the numerous advantages offered by the web services paradigm:

- Standard-based middleware technology.
- Business services high reusability level.
- Easy business legacy systems leverage.
- Integration between heterogeneous systems.

Due to these immediate benefits, most IT departments are implementing this technology with the high-priority objective of making them operable leaving aside – at least until later stages – the problems related to security. In general, the industry is still reticent to incorporate this technology due to the low understanding that they have of the security risks involved, and the false belief that they will have to make a costly reinvestment in their security infrastructures.

Web services as distributed, decentralized systems that provide well-defined services to certain (or not) predetermined clients, must be concerned with typical security problems common to the communication paradigm, through a compromised channel, between two or more parties.

2 Main Web Services Security Issues

The following section describes some of the major security issues that web services technologies must address:

Authentication: any web service that participates in an interaction may be required to provide authentication credentials by the other party. When certain service A makes a request for a service to a service B, the latter may require credentials along with a demonstration of its ownership (e.g.: a pair username/password or a X.509v3 certificate).

Authorization: Web services should include mechanisms that allow them to control the access to the services being offered. They should be able to determine who and how can do what and how on their resources.

Confidentiality: Keeping the information exchanged among web services nodes secret is another of the main properties that should be guaranteed in order to consider the channel secure. Confidentiality is achieved thanks to ciphering techniques

Integrity: This property guarantees that the information received by a web service remains the same as the information that was sent from the client. A simple checksum might offer integrity when accidental changes are made in the data.

Non-repudiation: In the web services world, it is necessary to be able to prove that a client utilized a service (requester non-repudiation) and that the service processed the client request (provider non-repudiation). This security issue is covered by means of digital signatures.

Availability: The need to take care of the availability aspects for preventing Denial-of-Service attacks or to arrange redundancy systems is a crucial point in web services technology. Above all, in those scenarios in which the services provide critical services: real-time services, Certification Revocation Lists services, etc.

End-to-end security: network topologies require end-to-end security to be maintained all across the intermediaries in the message's path. "When data is received and forwarded on by an intermediary beyond the transport layer, both the integrity of data and any security information that flows with it maybe lost. This forces any upstream message processors to rely on the security evaluations made by previous intermediaries and to completely trust their handling of the content of messages" [14].

Up to this point, we have briefly reviewed the typical security problems tightly related to distribute computing systems. Web services must address both these, inherited from the distributed computing classical scheme, and, in addition, those arising from the new threats created by its own nature. Some of these new threats are:

- Diversity and very high number of standard specifications that do not facilitate a clear vision of the problematic and its solutions.

- The current draft state in which majority of the security specifications are found.
- The Internet publication of a complete and well-documented interface to back-office data and company's business logic.
- New XML standard formats needed to structure the security data.
- Application-level, end-to-end and just-one-context-security communications.
- Interoperability of the requirements and on-line security elements
- Audit and automatic and intelligent contingency processes aimed at being machine-to-machine interactions not controlled by humans.
- A complex dependency network that can lead to the execution of a business process depending on an unknown web services number.
- On-line availability management in critical business processes.

The remainder of this article is divided into 5 parts. In the first one, a brief review of the core specifications that support the technology at hand is made. In the second section, core security web services specifications are explained, and unresolved issues not yet addressed by them are described. In the third and fourth parts, the main initiatives are introduced as well as the specifications related to the security that they are involved in. The fifth and last section, show how variety and, to a certain extent uncontrolled, specifications development and initiatives are already causing collisions among solutions to similar security problems.

3 Web Services Core Standards

In this section, we will take a look at the four fundamental standards involved in the creation of operational web services.
Figure 1 outlines the most important security specifications under development. They are grouped as following:

- Core: web services foundational specifications. These are the standards web services building are based on.
- Core Security: standards that provides the XML low-level security primitives such as ciphering and signing.
- WS-Security: family of specification developed by Microsoft and IBM which are under OASIS standardization process
- OASIS: security specifications developed by OASIS standards body.
- Liberty Alliance Project: represents the group of specifications developed in the Liberty Alliance Project.

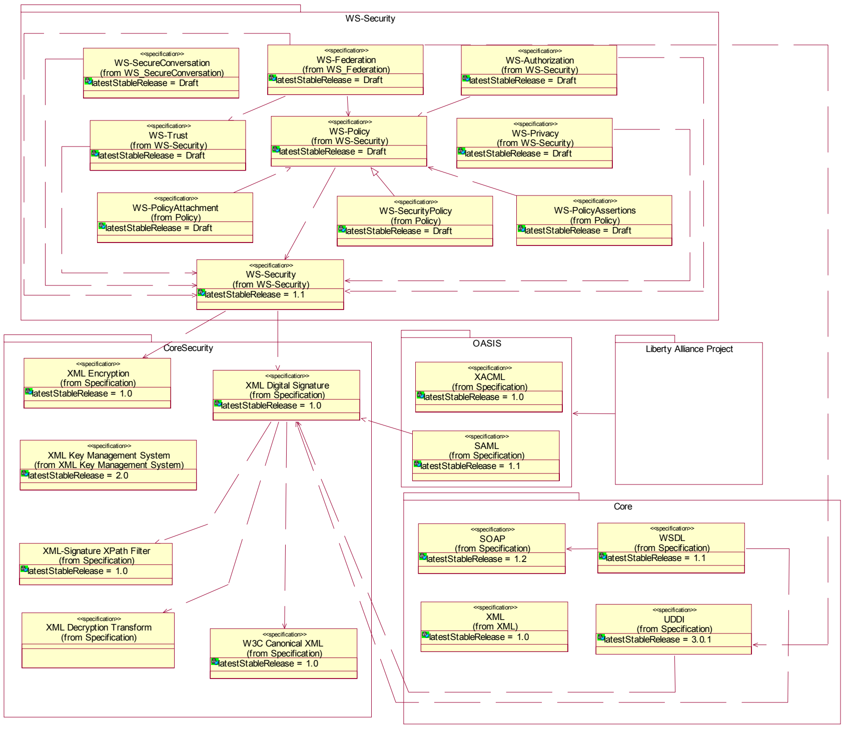

WS-Security
<<specification>> WS-SecureConversation (from WS_SecureConversation) latestStableRelease = Draft
<<specification>> WS-Federation (from WS_Federation) latestStableRelease = Draft
<<specification>> WS-Authorization (from WS-Security) latestStableRelease = Draft
<<specification>> WS-Trust (from WS-Security) latestStableRelease = Draft
<<specification>> WS-Policy (from WS-Security) latestStableRelease = Draft
<<specification>> WS-Privacy (from WS-Security) latestStableRelease = Draft
<<specification>> WS-PolicyAttachment (from Policy) latestStableRelease = Draft
<<specification>> WS-SecurityPolicy (from Policy) latestStableRelease = Draft
<<specification>> WS-PolicyAssertions (from Policy) latestStableRelease = Draft
<<specification>> WS-Security (from WS-Security) latestStableRelease = 1.1
CoreSecurity
<<specification>> XML Encryption (from Specification) latestStableRelease = 1.0
<<specification>> XML Digital Signature (from Specification) latestStableRelease = 1.0
<<specification>> XML Key Management System (from XML Key Management System) latestStableRelease = 2.0
XML-Signature XPath Filter (from Specification) latestStableRelease = 1.0
<<specification>> XML Decryption Transform (from Specification)
<<specification>> W3C Canonical XML (from Specification) latestStableRelease = 1.0
OASIS
<<specification>> XACML (from Specification) latestStableRelease = 1.0
<<specification>> SAML (from Specification) latestStableRelease = 1.1
Liberty Alliance Project
Core
<<specification>> SOAP (from Specification) latestStableRelease = 1.2
<<specification>> WSDL (from Specification) latestStableRelease = 1.1
<<specification>> XML (from XML) latestStableRelease = 1.0
<<specification>> UDDI (from Specification) latestStableRelease = 3.0.1

Fig. 1. Current security standards grouped by the organizations responsible for its standardization process

"Basic services, their descriptions, and basic operations (publication, discovery, selection, and binding) that produce or utilize such descriptions constitute the SOA foundation" [20]. Web services are built on an architecture SOA basis. In fact, web services architecture is a SOA architecture instantiation [7]. For that reason, the fundamental characteristics described by SOA are the ones that have initially headed the major efforts in the industry standards development process. The core web services specifications are: XML [24], SOAP [20], WSDL [16], and UDDI [1].

These specifications have been broadly adopted by the industry, constitute the basic building blocks on which web services are being designed and implemented. The bad news is that these four operative services specifications allow the creation of web services but they do not say anything about how to secure them. What's more, they themselves contain security questions that must be answered:

- *XML and SOAP*: these specifications do not say anything about how to obtain integrity, confidentiality and authenticity of the information that they represent and transport respectively.
- *UDDI and WSDL*: should answer questions like: Is the UDDI registry located in a trustworthy location? How can we be sure that the published data have not been maliciously manipulated? Was the data published by the business it is supposed to have been? Can we rely on the business that published the services? Are the

services available at any moment? Can we trust the transactions that are produced from the execution of the business services? As we can notice from all these questions, an in-depth analysis of the security problems that an UDDI and WSDL architecture implies is needed [5]. Despite all these drawbacks, these standards have evolved and matured and the industry has adopted and implemented most of them.

At this point, the main web services standardization initiatives are the World Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS). Both consortiums are trying to standardize their vision, security included, of what the web services should be and should contribute to the WWW world. This parallelism is causing the existence of specifications developed by both groups that resolve similar problems. As is expressed by IBM and Microsoft [10] "We note that other organizations such as the IETF and ebXML are tackling a related set of problems, and we are pleased there are already formal liaisons between the W3C XML Protocol Working group and its counterparts in both ebXML and IETF".

All the involved parts will have to make efforts to unify in the future with the purpose of integrating their visions and standards and thus, define a common and global framework.

4 Core Web Services Security Standards

The W3C consortium is responsible for the development of the following XML technology standards: XML Encryption; XML Digital Signature; XML Key Management System.

4.1 XML Encryption

W3C XML Encryption [15] is a Proposed since 2002. It provides a model for encryption, decryption and representation of: full XML documents; single XML elements (and all descendants) in an XML document; contents of an XML element (some or all of its children including all its descendants) in an XML document; and arbitrary binary content outside an XML document.

XML Encryption solves the problem of confidentiality of SOAP messages exchanged in web services. It describes the structure and syntaxes of the XML elements, which represent encrypted information and it provides rules for encrypting/decrypting an XML document (or parts of it).

The specification states that encrypted fragments of a document should be replaced by XML elements specifically defined in the recommendation. In order to recover the original information, a decryption process is also specified.

Web services use XML for delivering the necessary meta-information (SOAP headers) and the payload. As a result, XML Encryption can be used for encrypting/decrypting any fragment or logical part of a XML message. XML Encryption does not specify how to encrypt SOAP messages generated by web services. This task is delegated to higher-level specifications that would define rules for using this primitive within the information exchange context. XML Encryption also describes how to encrypt already encrypted content (super-encryption) and provides a mechanism for encrypting the keys used in the process.

Looking back at the beginning of this section, where a list is given of the datatypes that can be encrypted, we may miss the possibility of encrypting the tree nodes without having to encrypt full sub-trees. Basically, the solution would consist of extracting the wanted nodes from the original document, encrypt each of them and put them in an encrypted nodes pool. The recipient will get the modified document and the encrypted nodes pool, and it will be able to decrypt the nodes, which it is allowed to see and put them back in place inside the document [12].

One of the implicit security problems associated to this specification is the explicit declaration of the fragments that have been encrypted. According to the specification, information is encrypted and replaced by XML elements containing the result and so, analysis information attacks could be easily carried out on the output document.

Recursivity is also addressed, but no solution is given. Encrypted key A may need encrypted key B, but B may also need A. XML Encryption recommends the use of *ds:* namespace for these elements, which is where XML Digital Signature elements belong to, instead of providing a different namespace, more like the WS-Security family.

4.2 XML Digital Signature

XML Digital Signature [3] is a W3C recommendation since 2002, fruit of the joint work between W3C and the IETF. It defines how to digitally sign XML content and how to represent the resulting information according to an XML schema. Digital signatures grant information integrity and non-repudiation. Thus, for example, an entity cannot deny the authorship of a digitally signed bank transfer made through a web service.

According to the XML Digital Signature specification, a digital signature can be applied to any kind of digital content, including XML. It can be applied to the contents of one or more resources. Enveloped signatures and enveloping signatures exist. Both of them are applied over data contained within the same XML document that carries the digital signature. Detached signatures which sign digital content not contained within the same XML document also exist.

Signature creation and verification processes are defined by the specification. It is , like XML Encryption, technology independent, so additional mechanisms are needed to define how it will be applied to web services message exchange.

Applications using this specification combined with encryption must deal with some security related issues. The following rules are proposed:

- When the data are ciphered, any digest or signs on the data would have to be ciphered as well so that it is prepared to anticipate guessing plaintext attacks.
- Use XML Decryption Transform transformation during the digital signature verification process [2].

4.3 XML Key Management System

XML Key Management System [22]is a specification that has been subject to the W3C standardization process that proposes an information format as well as the necessary protocols to convert a Public-Key Infrastructure in a web service so that it will be able to: Register public/private key pairs; locate public keys; validate keys; revoke keys; and recover keys.

This way, the entire PKI is extended to the XML environment, thus allowing delegation of trustworthy decisions to specialized systems. XKMS is presented as the solution for the creation of a trustworthy service that offers all PKI subordinate services, but without resolving the inherent issues of the infrastructure:

- How can a Certification Authority's public key be known with total certainty? ¿Is the CA ascertained identity useful?
- Known issues with OIDs (Object Identifiers) for automatic processing and their explosive and continuing growth.
- Since the global public key infrastructure is lacking a single world-recognized certification authority, it is not clear how to equip the world in order to allow two systems (ex. web services) that know nothing of each other to establish a trustworthy relationship through a third party on the fly and with no previous off-line communication.

5 Web services Security: Standards and Security Issues Already Addressed

Now that we have reviewed the basic web services security standards and their related security, we turn to detail the emerging technology and specifications that are based on these standards.

First, we will briefly touch on the WS-* specifications, whose principal developers are IBM and Microsoft. Secondly and thirdly, we will talk about the SAML and XACML standards, which have already been delivered by the OASIS organization in their initial versions, and whose objective is how to present information and the security policy, respectively. Fourthly, we will briefly comment on the Liberty Alliance project, which is lead by Sun Microsystems, and fifthly and lastly, we will give a summary in matrix form showing all of the specifications that are covered in this paper, noting those that have been delivered and those that are still in draft form.

5.1 WS-Security Family Specifications

IBM and Microsoft, together with other major companies, have defined a web services security model that guarantees end-to-end communication security.

These companies are jointly elaborating a series of specifications that compose an architecture, termed by Microsoft as Global XML Web Services Architecture [8], that will lead the development in the web services industry so that different products can inter-operate within a secured context. The center of these specifications is composed by: WS-Addressing, WS-Coordination, WS-Inspection, WS-Policy, WS-Referral, WS-ReliableMessaging, WS-Routing, WS-AtomicTransaction, and WS-Security.

We will focus our attention to the last specification: WS-Security. IBM, Microsoft, and VeriSign developed and submitted to OASIS which is responsible of its standardization process. WS-Security [21] “describes enhancements to SOAP messaging to provide *quality of protection* through message integrity, message confidentiality, and single message authentication. These mechanisms can be used to accommodate a wide variety of security models and encryption technologies” .This is the specification on which some additional specifications (some with publicized versions) that cover all aspects of security in web services have based their definition. WS-Security is placed at the base of the security specification pile. Its purpose is to provide Quality of Protection to the integration, adding the following properties to communication and messages: message integrity, confidentiality and simple authentication of a message. WS-Security allows the easy incorporation of many existing security models such as PKI and Kerberos.

Other specifications that directly relate to security issues such as WS-SecurityPolicy, WS-Trust, WS-Privacy, WS-SecureConversation, WS-Authorization, and WS-Federation are being developed based on WS-Security.

In the protocol stack and right on top of WS-Security, we find the WS-Policy specifications (with its security attached WS-SecurityPolicy specification), WS-Trust and WS-Privacy.

WS-Trust is another specification deserving mention due to its similarity with XKMS. WS-Trust defines an XML schema as well as protocols that allow security tokens to be accessed, validated and exchanged. However, this is not a new problem since the XKMS specification already addresses it when the underlying security infrastructure is PKI. Therefore, if we wish to extend a PKI as web service, ¿which of the two standards should we use?

Another noteworthy specification is WS-Policy and its related specifications: WS-SecurityPolicy, WS-PolicyAssertions, WS-PolicyAttachment. These specifications define an XML syntax for defining web service policies (WS-Policy); a way to relate policies to XML elements, UDDI entries or WSDL descriptors; a combination of policy assertions of a general nature (WS-Policy-Assertions); and a combination of policy assertions of a security nature (WS-SecurityPolicy).

5.2 SAML

Secure Assertion Mark-up Language [11] is an "OASIS Open Standard" specification developed by OASIS and was delivered in its first version by 2002.

Basically, this specification defines a XML schema that allows trust assertions (authentication, authorization o attribute) representation in XML and request/response protocols to perform XML authentication, authorization and attribute assertion requests.

However, SAML has not yet resolved all the problems related to interoperable XML security-data transferences [13]. However it shows a significant progress. For instance, SAML does not solve how the authentication evidence itself is transferred. This issue has been addressed by WS-Security through its UsernameToken and BinarySecurityToken security tokens definition. In addition, SAML does not define the way to include SAML assertions within SOAP "wsse:Security" block headers (defined by WS-Security specification). In August 2002, WS-Security specification delivered the technical paper "The WS-Security Profile for XML-based Tokens" [23] in order to solve this matter.

5.3 XACML

XACML [19] is another OASIS specification since February 2003 and its main intention is to define an XML vocabulary for specifying the rules from which access control decisions can be enforced.

XACML defines these access control rules depending on the requester characteristics, communication protocol in use and the authentication mechanism used. XACML is very similar, as far as the security problem it solves, with the policy rules model and language defined by the previously studied WS-Policy family specifications. This coincidence is another example of the unification effort proof that an attempt will have to be made in the future to define a sole model and language policy-related in the web services world. XACML defines a service architecture that must be implemented by fully-fledged policy architectures:

5.4 Liberty Alliance Project

The Liberty Alliance Project [6], led by Sun Microsystems, and its purpose is to define a standard federation framework that allows services like Single Sign-On.

Thus, the intention is to define an authentication distributed system that allows intuitive and seamless business' interactions. As we can see, this purpose is the same as WS-Federation specification and Passport's .NET technology ones. Once again, this is another example of the previously so-called overlap problem in web services security solutions.

Table 1. Summary of the current web services standard development efforts grouped by topic.

Authentication	WS-Security, WS-Trust (draft), XKMS, SAML profiles (request/response protocol for obtaining SAML assertions), Liberty Alliance Project (SSO using extending SAML framework), WS-Federation (SSO) (draft)
Authorization	XACML (Policy-base authorization), WS-Authorization (draft)
Confidentiality	XML Encryption, WS-Security (draft)
Integrity	XML Digital Signature
Non-repudiation	XML Digital Signature, WS-Security
Security policies	WS-Policy + WS-SecurityPolicy (draft) , XACML
Trust authority	WS-Trust (draft) XKMS
Security contexts/ keys derivation	WS-SecureConversation
Delegation/Proxy	WS-Trust (draft), Delegation has not been fully addressed yet.
Privacy	WS-Privacy (draft)
Attribute mapping	??????
Reference security architecture	??????
Security methodology	??????

6 Issues to Be Solved

In spite of the amount of specifications that we have reviewed in this article, and summarized in Figure 1, there are a lot of unresolved security issues that will have to be addressed and standardized in the future:

1. A clear effort should exist from all entities involved in this technology in order to unify their criteria and solutions. The explosion of specifications and concepts is such that the learning curve may become unacceptable for the most of the IT projects. As it has been demonstrated during this article, questions like knowing whether the chosen solution is the best of all the possible ones or, if a solution has been chosen, it will be long-term supported by the major industry companies, are difficult to answer.

2. Another problem to be solved is attribute or role principal mapping among different systems. Coherent access control decisions will be difficult to be made when the same name of attributes or roles in both interacting web services are set. For instance, certain set of attributes assigned to user A in system Y may have a completely different meaning in other system B. System B should need to map the attributes provided by user A to its own attributes types in order to be able to make a coherent access decision. RBAC [4] together with a global attribute mapping agreement maybe the way to reach a successful solution.
3. Nowadays, a methodology that accomplishes and consider all the possible security issues and defines an organized development process that directs web services deployments in all expected (and unexpected) scenarios does not exist. This methodology should produce a distributed security framework. This framework would address all the necessary security primitives (authentication, security policy statements, confidentiality ...) and should be flexible enough as to allow primitive implementation solutions replacements without affecting the overall performance of the system. Thus, it should be able to define a framework where specialized security modules maybe plugged in. For instance, it should allow us to replace a WS-Trust security module for a XKMS module in a transparently way for the client. As a first approach, and inspired by SUN JMX architecture, we would design this framework by means of a security specialized *microkernel* creation in such a way. This *microkernel* would have a central component with not specific functionality beyond that as acting as socket where security modules can be plugged in. Every security module would plug in the socket by means of a well-known interface and would notice to the component about the security primitives it provides. Any client security request will be intercepted by the central component and then redirected to the correspondence security service. The response will be brokered by the central component as well.

7 Conclusion

In this article, we have reviewed the current web services security specification and initiatives and we have shown that its diversity is provoking an unclear vision of the problem and their solutions. In addition, unaddressed security issues have been stated overall and for each specification. The lack of a global standardization initiative is causing that overlapping solutions to similar problems are being put forward. This fact will require an extra effort in the future not only for the specifications to unify and make themselves interoperable but for industry to adopt and implement them. Therefore, solutions to topics like security policies, delegation, inter-business principal attributes mapping and privacy are not yet addressed by delivered and stable standards.

Acknowledgment

This research is part of the CALIPO project supported by Dirección General de Investigación of the Ministerio de Ciencia y Tecnología (TIC2003-07804-C05-03).

References

1. UDDI Version 3.0.1 - UDDI Spec Technical Committee Specification 14 October 2003. See http://uddi.org/pubs/uddi-v3.0.1-20031014.htm
2. Decryption Transform for XML Signature - W3C Recommendation 10 December 2002. See http://www.w3.org/TR/2002/REC-xmlenc-decrypt-20021210
3. XMLDsig. XML-Signature Syntax and Processing- W3C Recommendation 12 February 2002. See http://www.w3.org/TR/xmldsig-core/
4. RBAC. Role-based Access Control - Draft 4 April 2003. See http://csrc.nist.gov/rbac/rbac-std-ncits.pdf
5. Adams, C. and S. Boeyen (2002) UDDI and WSDL Extensions for Web Services: a security framework. Proceedings of the *ACM Workshop on XML Security*. Fairfax, VA, USA.
6. Liberty Alliance Project. Introduction to the Liberty Alliance Identity Architecture. See http://www.projectliberty.org/resources/whitepapers/LAP%20Identity%20Architecture%20Whitepaper%20Final.pdf
7. WSAS. Web Services Architecture Specification - WC3 Working Draft 8 August 2003. See http://www.w3.org/TR/2003/WD-ws-arch-20030808/
8. Box, D. (2002) Understanding GXA. See http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dngxa/html/gloxmlws500.asp
9. Casati, F., E. Shan, U. Dayal and M.-C. Shan (2003) Business-Oriented Management of Web Services. Communications of the ACM, Vol. 46, Nº 10, October 2003, pp. 25-28.
10. IBM and Microsoft. Web Services Framework. See http://www.w3.org/2001/03/WSWS-popa/paper51
11. SAML. Assertions and Protocol for the OASIS 2 Security Assertion Markup Language 3 (SAML) V1.1. See http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf
12. Geuer-Pollmann, C. (2002) XML Pool Encryption. Proceedings of the *Workshop on XML Security*. Fairfax, VA: ACM Press.
13. Harman, B., D.J. Flinn, K. Beznosov and S. Kawamoto (2003) *Mastering Web Services Security*. Wiley.
14. IBM and Microsoft. Security in a Web Services World: A Proposed Architecture and Roadmap - technical whitepaper 7 April 2002. See http://msdn.microsoft.com/ws-security/

15. XMLEnc. XML Encryption Syntax and Processing - W3C Recommendation 10 December 2002. See http://www.w3.org/TR/xmlenc-core/
16. WSDL. Web Services Description Language (WSDL) 1.1 - W3C Note 15 March 2001. See http://www.w3.org/TR/wsdl
17. Robert McMillan. IDC: Web services to enable $4.3B hardware market by 2007. IDG News Service (2003).See http://www.computerworld.com/hardwaretopics/hardware/story/0,10801,81496,00.html
18. O'Neill, M., P. Hallam-Baker, S.M. Cann, M. Shema, E. Simon, P.A. Watters and A. White (2003) Web Services Security. McGraw-Hill.
19. Papazoglou, M.P. and D. Georgakopoulo (2003) Service-Oriented Computing. Communications of the ACM, Vol. 46, Nº 10, October 2003, pp. 25-28.
20. SOAP. SOAP Version 1.2 Part 0: Primer. See http://www.w3.org/TR/2003/REC-soap12-part0-20030624/
21. WS-Security. Web Services Security (WS-Security) - Specification 5 April 2002. See http://www-106.ibm.com/developerworks/webservices/library/ws-secure
22. XKMS. XML Key Management Specification (XKMS) - W3C Note 30 March 2001. See http://www.w3.org/TR/xkms/
23. WS-Security Profile for XML-based Tokens - Specification 28 August 2002. See http://www-106.ibm.com/developerworks/webservices/library/ws-sectoken.html
24. W3C Extensible Markup Language (XML) 1.1 - W3C Recommendation 04 February 2004 (2004). See http://www.w3.org/TR/xml11

Secure Communications in Multi-Agent Systems Protecting KQML

Sierra J. M., Hernández J.C., Izquierdo A. and Ribagorda A.

IT security Group. University Carlos III of Madrid. Spain
sierra@inf.uc3m.es

Abstract. When multiagent systems use insecure networks their communications must be protected in the same way that any other applications that run over this type of channels. There is no doubt that multiagent systems expansion will be joined to the Internet technology, and for that reason our work tries to protect agents communications by a new security architecture and an extension of the KQML. Our security architecture has been designed to be installed over the RETSINA framework, which was specifically designed for an open system, such is the Internet. The core of our proposal is a SEcurity SubAgent Module, called SESAMO, which was expressly designed to easily interact with the RETSINA components. The protection is based a public key infrastructure that, in addition to an extension of KQML, will supply authentication, non-repudiation, integrity and confidentiality services to agent communications.

1 Introduction

KQML, *Knowledge Query and Manipulation Language*, permits autonomous and asynchronous agents share their knowledge and work cooperatively for solving problems. The possibilities of Multi-Agent Systems (MAS) increase considerably if they use the Internet. But it is necessary to adapt the KQML to this open environment, supplying to the agents security services such are confidentiality, integrity, authentication and non-repudiation.

First aim of this security architecture is to effortlessly coexist with other multiagent systems. Our proposal is designed to work over the RETSINA framework. The core of our architecture is the SESAMO module (SEcure SubAgent MOdule). This module supplies cryptographic capabilities to RETSINA Task Agents, permitting them to establish secure communications with others. The SESAMO module can be installed into a Host Agent or also allows that several agents (agents connected by a private network or installed into the same machine) to share a single SESAMO, we called that option Shared SESAMO. We also describe some other functions that can be developed by SESAMO because its design can be used as a communications security gateway between groups of agents.

Agents that want to interact directly with their parties can bypass our architecture. A common situation could be that Task Agents just use the SESAMO when the remote agent is asking for a secure connection, or when they want to establish this type of connections with others. In the rest of situations they will communicate openly with KQML. The SESAMO modules will communicate using a security extension of KQML that we have designed. This extension is called KQML-SE and is composed by three new performatives:

1. Cryptographic Capabilities Negotiation.
2. Both Parties Authentication messages.
3. Encapsulated KQML messages.

Our scheme is based on public key cryptography, and obviously this environment needs the existence of a Public Key Infrastructure. This PKI will be used for the authentication of agents and hence the architecture security relies on it. We have designed another extension of KQML that provides the performatives needed for the creation, renewal and cancellation of certificates, and also for the maintenance of Certificate Revocation Lists (usually noted as CRL's). All these topics will not be treated in this paper because we would need some more extra space to describe its behaviour and management. However, we are aware about the essential role that PKI plays in our architecture.

2 Security Architecture

When we start this work we tried to develop architecture easy to place in a Multi-Agent System. This architecture should introduce the minimum number of changes, enabling an agent to integrate security services with no modifications on its basic architecture. In this paper we concentrate our work in the well-know architecture called RETSINA. In our proposal each agent (from now on, Host Agent) has another sub-agent associated, called SESAMO (SEcurity Sub-Agent MOdule). SESAMO module is in charge of dealing with all the KQML messages sent or received by the Host Agent.

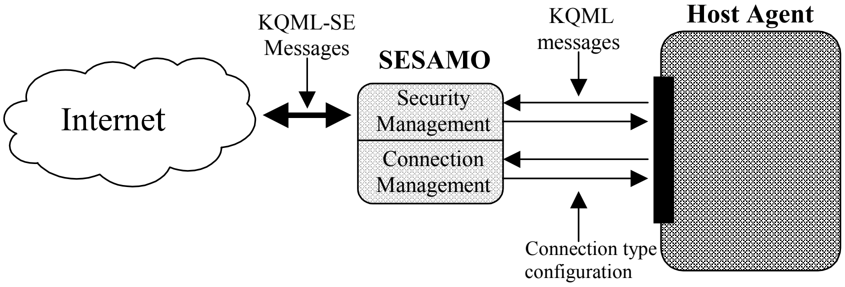


Fig. 1. : KQML-SE architecture based on the SESAMO

The SESAMO only manages performatives defined by our new ontology PKCertificate. SESAMO has implemented all the cryptographic capabilities that the Host Agent does not have; in this way the SESAMO will provide all the required security services (authentication, non-repudiation, confidentiality and integrity). On the other hand the agent will be working with no change, security services will be applied by the SESAMO transparently to the Host Agent. The Host Agent only must indicate to the SESAMO which messages will need protection.

The SESAMO will be divided into two parts, *Security Management* part, which will be in charge of protecting the KQML messages exchanged between the Host Agent and the other agent. And the second part, *Connection Management*, which will be use by the Host Agent to specify to the SESAMO the connection features. The Host Agent must indicate to the SESAMO what to do in case that other remote agent does not have his own SESAMO. For example a Host Agent could block any message that is not authenticated, in this case the SESAMO will be protecting his agent from non-authenticated communications just keeping away from messages that does not contains the PKCertificate ontology. The SESAMO could filter communications, for example rejecting messages with certain source address, size, etc..

Another functionality that can be included into the SESAMO is a Connection Features Database. Into this database will be stored the connection features of a pair of agents (the Host Agent and the remote Agent). Using this database will be skipped the main work of the Connection Management part of the SESAMO. This database will be particularly useful when a Shared SESAMO is used (see Sharing a SESAMO Agent section)

An important advantage of our design is that the communication is possible in any case. If one of the parties does not have a SESAMO module, the other SESAMO can bypass the common KQML messages (without the PKCertificate ontology) and permits to establish the communication.

Table 1. : Optional functionalities of SESAMO module

Function	Type
Encapsulating (KQML ↔ KQML PKCertificate).	Obligatory
Strong Authentication of both agents	Obligatory
Digital signature of KQML messages	Obligatory
Shared SESAMO	Optional
KQML messages filtering	Optional
Connection features Database	Optional

3 KQML Security Extension. KQML-SE

The KQML security extension consists on four new performatives. These performatives will be used in three steps. The first one is the negotiation of the Cryptographic capabilities. The new performative designed is ***negotiation,*** which enables to negotiate: Certification Authorities, Digital Signature Algorithm, Cipher Algorithm and Digest Algorithm.

Once agents know the cryptographic capabilities one each other, it begins the second step where is accomplished the authentication of the parties and the establishment of a shared secret key (this key will be based on information supplied by both parties). The performatives that support this second step are Auth-link and Auth-challenge.

Finally, once the parties are properly authenticated and a ciphering key is established, it is possible to set up a secure channel. The performative is called ***Auth-private***.

Next subsection describes the new parameters involved.

3.1 New parameters

3.1.1 :certificateCA(<Certification Authority 1><Certification Authority 2>,...)

Where the argument <Certification Authority X> is the identification associated to one Certification authority (Thawte, VeriSign). This parameter is an enumeration of the different certification authorities supported by the agent. It is a parameter of the Negotiation performative.

3.1.2 :certificate (<Certification Authority><the certificate>)

The second argument is the certificate, which is a public key signed by a certification authority (indicated in the first argument).

3.1.3 :connection-id (<NONCE_X><NONCE_Y >)

This parameter is used in the Auth-private performative and identifies a previous negotiation.

3.1.4 :auth-key(<boole><key-type><SEED>)

First argument is a Boolean value. If this value is TRUE the session key must be changed. The new key to use will be the result of a hash function calculated over the concatenation of SEED, NONCE_X and NONCE_Y (the resulting digest could need to be adapted to the encryption algorithm key size. This operation is done to ensure that the new key depends from both parties.

If the first argument is FALSE, next arguments will be ignored because the agent is signifying that, for the moment, the session key is valid. Any party of the communication could indicate a key change when it considers appropriate. This parameter is included into Auth-private and Auth-challenge performatives.

3.1.5 :signature(<<Key_ID><information signed>)

The second argument represents certain information digitally signed by the agent. The first argument identifies the public key that must be used to check the signature. It is included into Auth-link, Auth-challenge and Auth-private performatives and provides Authentication and Non-Repudiation of the messages.

3.1.6 :algSecretKey (<alg1><alg2>...<algN>)

It indicates the different symmetric cipher algorithms supported by the agent that sends this parameter into a Negotiation performative. Also it is used in Auth-link performative, in this case an agent is indicating to its party the selection of certain algorithm.

3.1.7 :algSignType (<alg1><alg2>...<algN>)

This parameter indicates the different digital signature algorithms supported by the agent that sends the message. Can be used into a Negotiation performative and in Auth-link performative but in this case it is signifying the selected algorithm

3.1.8 :algDigestType (<alg1><alg2>...<algN>)

This parameter indicates the different digest algorithms supported by the agent that sends the message. Can be used into a Negotiation performative and also it can appears in a Auth-link performative but in this case just with the algorithm selected

3.1.9 :mySESAMO (<SESAMO_ID><protocol><address>)

It is used into the Negotiation performative and indicates what SESAMO agent is being used for protecting messages.

3.2 New Performatives

Into this section we will present the new performatives defined into KQML-SE. Those performatives should be added to those included into RETSINA.

Table 2. New performatives for KQML-SE

Performatives	Meaning
Auth-link	Request of secure communication
Auth-challenge	Acknowledge for Auth-link request
Negotiation	Cryptographic capabilities negotiation
Auth-private	KQML messages ciphered and encapsulated

In the following lines we describe the content of all these new performatives:

Name: Negotiation.
Description: Cryptographic capabilities negotiation between two SESAMO's
Additional paramenters:
:mySESAMO (Only has to be used when a Shared SESAMO is used)
Ontology: PKCertificate
KQML Description:
Negotiation:

```
            : sender <A>
: receiver <B>
```

: certificateCA<VeriSign><Thawte><Internal Domain >.....>
: mySESAMO <<X<tcpip><X@domain.com>>
: algDigestType<<MD5><MD4><SHA>...>
: algSecretKey<<DES><RC2>...>
: algSignType<<RSA><DSA>>
: ontology <PKCertificate>

Name: Auth-link
Description: Solicitud de comunicación segura.
Optional parameters:
:peer-address, only used when a shared SESAMO is used.
:algDigestType, list of digest algorithms supported.
:algSecretKey, list of symmetric algorithms supported.
Ontology: PKCertificate
KQML Description:

```
Auth-link
        :sender <A>
        :receiver<B>
        :reply-with<expresion>
:algDigestTypeType<MD5>
:algSecretKey<DES>
:algSignType<RSA>
        :certificate <<VeriSign><Certificate_A>>
:signature<<A_KEY> <Signature of (NONCE_A & A & B )>>
        :content: <NONCE_A>
```

Name: Auth-challenge
Description: Acknowledge for Auth-link request
Ontology: PKCertificate
KQML Description:

```
Auth-challenge:
        :sender <B>
        :receiver<A>
        :in-reply-to<expresion1>
        :reply-with<expresion2>
        :certificate <<VeriSign><CertificateB>>>
:auth-key<<T><DES><SEED ciphered by the public key of A>>
:signature<<KEY_B><Signature of (NONCE_A & NONCE_B & SESSION_KEY)>>
:content: <NONCE_B>
```

Name: Auth-private
Description: KQML message ciphered and encapsulated.
Ontology: PKCertificate
KQML description:

```
Auth-private:
        :sender <A>
        :receiver<B>
        :in-reply-to<expresion1>
        :reply-with<expresion2>
        :connection-id<<NONCE_A><NONCE_B>>
:auth-key< <FALSE> <><> >
:signature<<A_KEY><Signature of(KQML message)>>
        :content <KQML message ciphered with SESSION_KEY[1]>
```

4 Sharing a SESAMO Agent

The SESAMO module can be shared among more than one Host Agent. In this way all the KQML-SE messages must be sent to the SESAMO and the rest of non-protected KQML messages can be sent by each agent itself. The advantages of this approach are very interesting, because using a shared SESAMO the agents can be working with KQML in the same way that they are already working. And, in case they need a protected communication with other agent, they will use the SESAMO for establishing a secure channel.

The SESAMO needs to implement extra software for the management of several connections at the same time. This software will be able of storing the connection features of every agent into the commented Connection Features Database. The shared SESAMO is a distributed scheme. In this way any agent with authorization can use a shared SESAMO. However this system has been designed for being used in a community of agents connected by a private network (or agents that are running into the same machine). The utilization of Shared SESAMO through insecure networks needs additional protections that probably imply an excessive cost and more complicated management.

[1] SESSION_KEY is the digest of NONCE_A, NONCE_B AND SEED. The digest funcition used is the agreed in the negotiation process.

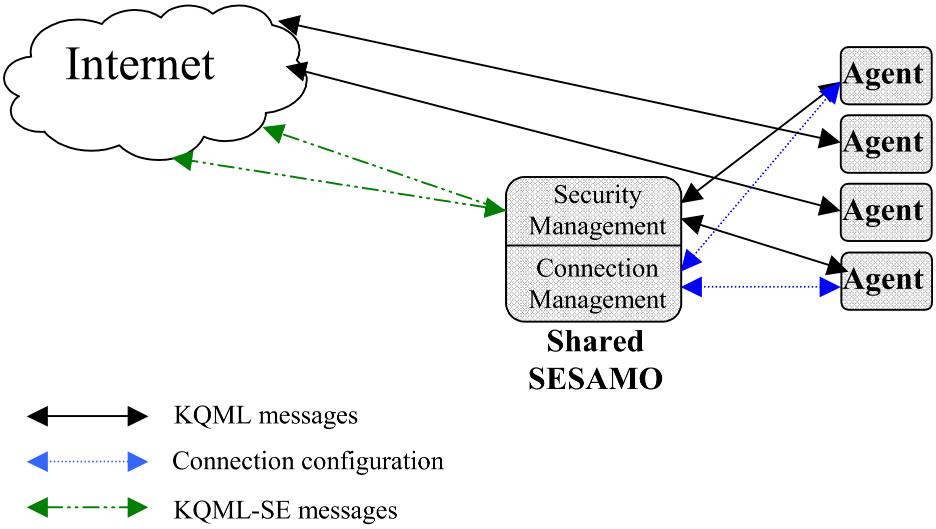


Fig. 2. KQML-SE architecture based on a Shared SESAMO

5 Conclusions

The expansion of the Internet has important implications for the MultiAgent Systems. However the Internet features must be taken into account because all of them will be inherit by the applications that run over it. With this document we tried to outline the security architecture for protecting KQML communications. We are aware that the implementation of our proposal is not completely described here, but this topic is already open in our research group and we will report new contributions.

KQML-SE and SESAMO are part of a complete system with the objective of providing secure communications. Foundations of this system are the public key cryptography and the implementation of a Public Key Infrastructure. We have developed an architecture for the management of PKI in a multiagent system, we have called this module SPA –Security Proxy Agent-. Our work is based on the contributions of Sycara about Security Agents.

Further research on this topic can be associated to the utilization of IPsec framework for the communication of the agents (the new IP Security Protocol that supplies authentication, integrity and confidentiality of IP packets). In this way the SESAMO module can be substituted by an IPv6 implementation on the agents. However, if we use IP security, it does not replace the task of the mentioned SPA module. The combination of the SPA and IPsec protocol will be a next step for our future research works.

6 References

[1] W. Timothy Polk, Donna F. Dodson, etc, Public *Key* Infrastructure: From *Theory to* Implementation, http://csrc.ncsl.nist.gov/pki/panel/overview.html, NIST

[2] Tim Finin, Yannis Labrou, and James Mayfield, *KQML as* an agent communicatíon language, in Jeff Bradshaw (Ed.), "Software Agents", MIT Press, Cambridge (1997).

[3] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, Tatu Ylonen, Simple *Public Key Certificate,* http://www.clark.net/pub/cme/spki.txt

[4] Ronald L. Rivest, Butler Lampson, *SDSI* - A Simple *Distributed Security* Infrastructu,re, http://theory.lcs.mit.edu/ cis/sdsi.html

[5] Bruce Schneier,Applied Cryptography, Second Edition, John Wiley and Sons, Inc., 1996.

[6] Matt Blaze, Joan Feigenbaum, Jack Lacy, *Decentralized Trust* management, In Proceedings 1996 IEEE Symposium on Security and Privacy, May, 1996.

[7] Sycara, K., Decker, K, Pannu, A., Williamson, M and Zeng, D., Distributed Intelligent Agents. IEEE Expert, pp.36-45, December 1996.

[8] Tim Finin, James Mayfield, Chelliah Thirunavukkarasu, Secret Agents - A Security Architecture for the KAML Agent Communication Language, CIKM'95 Intelligent Information Agents Workshop, Baltimore, December 1995.

[9] Qi He, Katia P.Sycara. Personal Security Agent: KQML-Based PKI. October 1997.

Hiding Traversal of Tree Structured Data from Untrusted Data Stores *

Ping Lin and K. Selçuk Candan

Department of Computer Sciences and Engineering
Arizona State University
Tempe, AZ. 85287

Abstract. With the increasing use of web services, many new challenges concerning data security are becoming critical. Especially in mobile services, where clients are generally thin in terms of computation power and storage space, a remote server can be outsourced for the computation or can act as a data store. Unfortunately, such a data store may not always be trustworthy and clients with sensitive data and queries may want to be protected from malicious attacks. In this paper, we present a technique to hide tree structured data from potentially malicious data stores, while allowing clients to traverse the data to locate an object of interest without leaking information to the data store. The two motivating applications for this approach are hiding (1) tree-like XML data as well as XML queries that are in the form of tree-paths, and (2) tree-structured indexes and queries executed on such data structures. We show that this task is achievable through a one-server protocol which introduces only a limited and adjustable communication overhead. This is especially essential in low bandwidth (such as wireless) distributed environments. The proposed protocol has desirable communication and concurrency performance as demonstrated by the experiments we have conducted.

Keywords: XML, content privacy, access privacy.

1 Introduction

In web and mobile computing, clients usually do not have sufficient computation power or memory and they need remote servers to do the computation or store data for them. Publishing data on remote servers helps improve data availability and system scalability, reducing clients' burden of managing data. With their computation power and large memory, such remote servers are called data stores or oracles. Typically, these data stores can not be fully trusted, for they may be malicious and can make illegal use of information stored on them to gain profits. Clients with sensitive data (e.g., personal identifiable data) may require that their data be protected from such data storage oracles. This leads to encrypted database research [1, 2], in which sensitive data is encrypted, so the content is hidden from the database. It is defined as *content privacy* [3].

* This work is supported by the AFOSR grant #F49620-00-1-0063 P0003.

Sometimes not only the data outsourced to a data store, but also queries are of value and a malicious data store can make use of such information for its own benefits. This privacy is defined as *access privacy* [3]. Typical scenarios demanding access privacy include:

- A mineral company wants to hide the locations to be explored when retrieving relevant maps from the IT'department map database.
- In a stock database, the kind of stock a user is retrieving is sensitive and needs to be kept private [4].

This leads to private information retrieval [4] research, which studies how to let users retrieve information from database without leaking (even to the server) the location of the retrieved data item.

Tree structure is a very important data structure and tree-structured data shows itself in many application domains. In this paper, we address outsourcing and hiding of tree-structured data and queries on this data. For this work, we have two motivating applications: (1) hiding XML data that is stored in the form of trees and XML queries in the form of tree paths; (2) hiding tree indexed data and queries for the data.

In this paper, we concentrate on hiding tree structured data and traversal of trees from oracles. Noticing that existing private information retrieval techniques require either heavy replication of the database onto multiple non-communicating servers or large communication costs [4], we give an one-server *tree-traversal* protocol that provides a balance between the communication cost and security requirements. To protect the client from the malicious data store, some tasks (such as traversing the tree-structures) are delegated to client.

This paper: In Section 2 we present a general overview of the framework and the outline of the hidden data access. In Section 3, we discuss how redundancy enables oblivious traversal of a tree structure. In Section 4, we address the underlying technical challenges and provide traversal algorithm. In Section 5 we give a quantitative analysis of the protocol and discuss how to tune the various system and security parameters to optimize the performance. We implement the protocol and analyze experiment results in Section 6. Section 7 discusses the amount of security the protocol can achieve and suggests ways to improve the security of the protocol in the future. Finally, we conclude in Section 8.

2 Overview of the Hiding Framework

In this section, we first give a general overview of the hiding framework. We, then, provide an outline of the proposed hidden data retrieval protocol.

There are three types of entities with different roles in the system: data owners, licensed users, and a data store (oracle). The data owners and licensed users are thin clients (as explained before). A data owner has the right to publish its data on the oracle, and a licensed user has the permission granted by some data owner to retrieve information from the data owner's data storage space in the oracle. The oracle manages data storage spaces, where data and tree structures are stored in a hidden way.

Clients run data encryption algorithms, have initial secret keys for decryption. Encryption algorithms are used to encrypt data and tree structures before sending them to the oracle to ensure that the content of data and the data structure are hidden from the oracle. If clients are accessing an outsourced index tree, they have point- or range-queries. If they are accessing outsourced XML trees, they have query patterns. Query patterns are used to traverse a tree structure along paths described by some regular-like expressions. These tasks are accomplished efficiently by ”thin” clients with the help of specialized embedded hardware, such as smartcards, distributed to licensed user by data owners. Smartcards have been used a lot in mobile computing. They are relatively cheap, costing no more than several dollars. Such embedded hardware also helps in solving secret key distribution problem, i.e. by distributing smartcards that contain secret keys, a data owner distributes keys to licensed users[5].

Every time the data owner wants to insert new data into the tree structure or delete a data item from it, the owner

1. encrypts the data with a secret key,
2. walks the index structure in an oblivious manner so that the traversal path is hidden to the data store
3. locates the node of interest (either for insertion or deletion),
4. updates the tree structure by inserting or deleting encrypted index or data nodes in proper positions in the tree, in an oblivious way with respect to the data store.

By walking or updating the tree structure in an oblivious way with respect to the data store, we mean minimizing the leakage of information about the data and the tree structure as much as possible; the details of how to walk and update tree-structures in an oblivious way is described in Section 4.

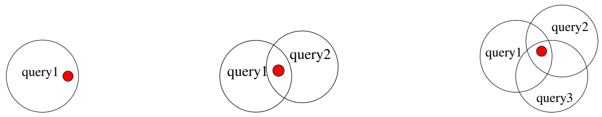
Client traversal of the tree for retrieving information is similar to update as in order to prevent the database server from differentiating between read and write operations, a read operation is always implemented as a read followed by a writing of the contents back.

3 Oblivious Traversal of the Tree Structure

It is obvious to hide the content of the nodes of a tree structure by encrypting them before they are passed to the data store. Consequently their content is already hidden from a malicious store. However, if a client traverses the tree structure in a plain way, the relationships between nodes in the tree, therefore the tree-structure as well as the user’s query, are revealed. We propose two adjustable techniques to achieve oblivious traversal of tree structures: *access redundancy* and *node swapping*.

Access Redundancy: Access redundancy requires that each time a client accesses a node, instead of simply retrieving that particular node, it asks from the server a set of randomly selected $m - 1$ nodes in addition to the target node. Consequently, the probability with which the data store will guess the intended node is $\frac{1}{m}$. m is a security parameter that is adjustable. We discuss how to choose the value of m in Section 5. We define this set the *redundancy set* of the target node.

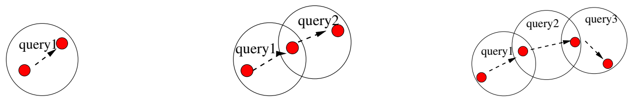
The problem with redundancy sets, on the other hand, is that their repeated use can leak information about the target node. For example, if the root node’s address is fixed,



(a) After first access (b) After the second access (c) After the third access

Fig. 1. Leakage of the position of root node of index as a result of repeated accesses

then multiple access requests for the root node reveal its position (despite the use of redundancy) since the root is always in the first `redundancy set` any client asks. By intersecting all the `redundancy sets`, the data store can learn the root node. The situation is depicted in Figure 1. If the root is revealed, the risk that its children may be exposed is high, and so is the case with the whole tree structure.



(a) After the first access (b)After the second access (c) After the third access

Fig. 2. The movement of a node

Node Swapping: Consequently, in order to prevent the server from using an attack based on intersecting repeated or related requests, we have to move nodes each time they are accessed. Preferably, the move should have minimal impact on the tree structure and should not leak information about where a given node is moved to. To achieve this, each time a client needs to access a node from the server, it asks from the server a `redundancy set` consisting of m nodes that includes at least one empty node along with the target node. The client then

```
1. decodes the target,    2. swaps it with the empty,
3. re-encrypt the redundancy set and writes them back
```

Figure 2 shows how this approach prevents information leakage: Figure 2(a) shows that after the first access, the position of the target node is moved (the arrow shows the node's movement). Figure 2(b) and 2(c) show that after the second and the third accesses, the position of the target node is moved again. As shown in Figure 2, during the course of an access, the oracle has the chance to know the position of the node only if the `redundancy set` for the access has little intersection with the set of the previous access so that the position where the node moved to after the previous access is revealed. But since the node moves again once the nodes are written back after the access, such leakage is of no use to the server. In this way, the possible position of the target node is randomly distributed in the data storage space and thus the repeated-access-attack is avoided.

Node swapping requires *re-encryption* of nodes before they are re-written to the server. Re-encryption should employ a new encryption scheme/key, the reason is as follows: if the same encryption scheme is used, by comparing the content of nodes in the `redundancy set` after rewriting with their original content, the server can easily identify the new position of the node. This means that a client has to identify how each node is encrypted. We achieve this by adding a new field which contains the secret key for that particular node. This field is always encrypted using a single/fixed secret key.This way, the client can decrypt this field to learn how to decrypt the rest of the node.

4 Hidden Tree Traversal Algorithm

To implement oblivious traversal of tree structure, some critical issues have to be solved:

- After moving one node, in order to maintain the integrity of the tree structure, the parent's pointer to this node has to be updated accordingly. How can this be performed without revealing parent-child relationships on the tree structure?
- How to keep consistency of a tree structure when there are many clients access it concurrently?
- How can we choose the values of various system parameters, such as the amount of redundancy m?

In this section, we provide techniques to address the first two of these challenges, and provide hidden retrieval algorithms based on them and the underlying protocol. In Section 5, we will discuss the choice of system parameters in greater detail.

Maintaining Parent/Child Relationships: As to the challenge of maintaining node/parent-node relationships after node swapping, we propose the following solution: find the empty node to be swapped with the child node and update the parent node correspondingly before actually moving the child node. This way, parents are always updated considering the future locations of their children.

Concurrency Control without Deadlocks: The proposed protocol will be applied to web-based mobile computing environments with large number of clients. In order to keep consistency of the tree structure with many clients accessing tree structures simultaneously, proper concurrency control must be used at server's side. There has been intensive study about index locking so that maximum concurrency is achieved with the integrity of tree structure preserved [6–8]. Since there is no pure read operation in the scheme (each node, after being read, should be written back), only exclusive locks are needed. To prevent deadlocks, we organize nodes in a data owner's data storage space into d levels.Each level of a data owner's data storage space requires an empty node list to maintain empty nodes at this level. Client always asks for locks of parent level nodes before asking for locks of child level nodes, and it always asks for locks of nodes belonging to the same level in some predetermined order (e.g. in the order of ascending node ids). In this way, all nodes in a data owner's data storage area are accessed by all clients in a fixed predetermined order. This ensures that circular waits can not occur, hence deadlocks are prevented.

In Figure 3, we provide the pseudo code of the oblivious traversal algorithm. The time complexity for this algorithm is O($d \times m$), with d denoting the depth of tree storage space and m denoting the `redundancy set` size, and the space complexity for it is O(m).

```
[Oblivious traversal algorithm]
Input: feature values of target data and the identifier of the data owner.
Output: pointer to the node that contains the data if there exists one; or null pointer.

  1. lock and fetch the fixed public entry node to the data store, let it be PARENT, find the
     root, let it be CURRENT.
  2. select a redundancy set for the CURRENT, lock nodes in the set, let the empty
     node in the set be EMPTY.
  3. update the PARENT's pointer to refer to the EMPTY, release locks on the PARENT
     level.
  4. swap the CURRENT with the EMPTY.
  5. if CURRENT contains the data, return CURRENT
     else
     let CURRENT be PARENT, find the child node to be traversed next, let it be CUR-
     RENT, repeat 2,3,4,5.
```

Fig. 3. Oblivious traversal algorithm

5 Identifying Appropriate Values for the System Parameters: Hiding a Single Query

Choosing the appropriate design parameter values for a hiding system depends on various system constraints, including the acceptable communication cost and the required degree of hiding. Let us model a data owner's data storage space as d levels. Suppose the tree structure is an l-level tree. Then, the following parameters and constraints have to be considered:

- the maximum probability, δ, for the server to be able to find the actual node that the client is asking from a `redundancy set`. We have: $\frac{1}{m} \leq \delta$.
- the maximum probability, λ, for the server to find the path along which a client walks the tree structure. We have: $\frac{1}{m^l} \leq \lambda$.
  We emphasize here that although it is easy for the data store to guess the target node from the `redundancy set` if m is small, it becomes much harder to guess the parent-child relations between sequential node accesses. And the probability to discover a path is reduced exponentially with the increase of length of the path, hence should be slim even with a small value of m.
- the total communication cost ε clients are allowed to make for each data retrieval. We have: $((read(m) + write(m)) \times l \leq \varepsilon$, here $read(m)/write(m)$ denotes communication cost to read/write m nodes from the server.
- a node may contain multiple data points. We denote the node size, i.e. the number of data points a node is able to contain, as s. Value of s can be determined by considering the following:
  Let c denote the function of one round-trip communication cost for data points to be received from and sent to the server, e and d denote the encryption and decryption

cost function , w and r denote the write and read cost function. Theoretically, they are linear functions. Then :

```
total_cost_for_data_retrieval
=   tree_depth * m *(communication + decryption + encryption +
                         read + write cost_per_node)
=   l * m *( c(s) + d(s) + e(s) + r(s) + w(s) );
```

As node size s increases, tree depth l decreases while costs per node increases. If all other parameters are known, we can calculate *optimal* node size to minimize the total cost. However, as s increases, the probability for the data store to find a path, which is $\frac{1}{m^l}$, increases. Therefore, the value of s should be carefully chosen to ensure that security requirement is satisfied and the total cost is minimized as much as possible.

Note that most of the above constraints are linear, and an appropriate parameter setting can be easily identified using efficient algorithms.

6 Experiment Results

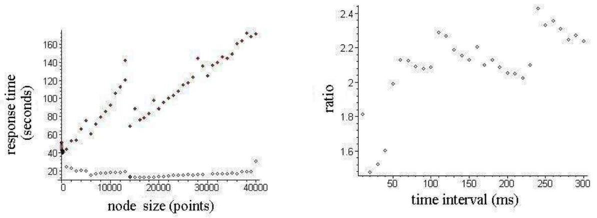
To validate the protocol, we simulated the protocol and conducted some experiments to test the protocol. The computing environment consisted of a Linux server acting as a data store and a 1.0Ghz/256M laptop generating client requests. They were connected via a Wireless LAN system. We implemented a 2 dimensional k-d tree as the index structure due to its simplicity. This simple structure enables us to observe experiment results more effectively.

In the paper, we do not experiment with range queries as we focus on path traversal. We point out that using this protocol, range queries can be implemented as multiple path traversals without deadlocks. We generated 40000 data points that were uniformly distributed in the region (0,0) to (1000000, 1000000), and stored them into a data storage space with capacity 30000 nodes. The size of `redundancy set`, m, is set to 8.

Response time and node size We executed a set of experiments to show the relationship between node size and response time, i.e., the time between a client sending a data retrieval request and getting the response.

Figure 4(a) shows the experiment result. In this figure, there are two sets of results. The dark points denote the results of experiments with encryption/decryption implemented by software. This set of results shows that when node size is set to around 50 data points, the minimum response time (about 38s), is achieved. This phenomenon verifies the theoretic observation that there must exist an *optimal* node size (Section 5). Considering the probability for the malicious server to find the path (we denote it as path probability, which is a function of page size, $\frac{1}{m^{log(\frac{num}{s})}}$, here m is the redundancy parameter, num is the total number of data points stored, s denotes node size.), suitable node size can be chosen to satisfy security requirements and minimize response time.

The set of white points depicts experiments with efficient hardware encryption/decryption. From the result, we found that encryption and decryption constitute heavy cost and with assistant hardware, response time can be greatly reduced to about 8s.



(a) response time and node size

(b) ratio of sequential process and concurrency control

Fig. 4. Experiment result

To compare our protocol with one-server Private Information Retrieval (PIR) technique [4], we also simulated PIR by transferring the whole database to a client. The simulation was conducted in the same computing environment (same linux sever, same laptop, same Wireless LAN connection). It takes about 3643s to finish transferring. We can claim that our protocol is much more efficient.

Another interesting phenomenon we observe from Figure 4(a) is that although the two sets of points have big difference in their values, they have similar zigzag pattern. This shows that the discontinues and sharp varieties in response time values are mainly determined by other costs (communication cost $c(s)$, write cost $w(s)$, read cost $r(s)$) than encyption/decryption (Section 5).

We also notice that response time for the set of black points has a strong tendency to increase with the node size, while it does very slightly for white points. This can be explained by the significant parts encryption/decryption play in the total cost and their linear increase with the node size (Section 5).

Furthermore, we conducted a set of experiments to show the effect of concurrency control. In this set of experiments, 50 retrieval requests for independently selected random data points were launched out one by one at varying frequency from every 10ms to every 300ms. In the experiment results, we found no deadlocks. We also found that the total time to finish all the requests was much less than letting the server process those retrievals sequentially. To give a sample result, when requests were launched out every 20ms, the total time required to finish them was 734.8s, and the time to process them sequentially was 1442.9s. Figure 4(b) gives the ratio of the time required to process sequentially and the time required by our protocol with concurrency control. We can see that the ratio is about 2. This means we gain 100% saving with the concurrency control. Figure 4(b) also shows that this ratio increases with the time interval. This is consistent with the common knowledge that the efficiency of the data store reduces with more clients accessing trees at the same time.

7 Future Work on Hiding Correlated Queries

The protocol should be able to protect queries and tree data structure from a polynomial time server. To study the security guarantee the protocol provides, suppose that the server keeps a history of all `redundancy sets` users retrieved, and the server tries to infer about queries and data by statistic analysis of the history. We define each `redundancy set` a **call**, and the history a **view** of the server. The amount of security is defined as:

1. For any two different queries Q_1 and Q_2 posed in the view, the distribution of their sequences of calls are indistinguishable in polynomial-time.
2. For any two queries Q_1 and Q_2 posed in the view, it is hard to tell if they are identical or not by observing their sequences of calls.

If the data storage space is randomly initialized, queries are uniformly posed, tree nodes will always be uniformly distributed in each layer of the data storage space. So for two different queries, if their query path lengths are equal, the distribution of their sequences of calls are identical, hence indistinguishable in polynomial-time; if their query path lengths are not equal, clients can execute dummy calls at deeper levels to always make the same number of calls. We are currently studying how to improve the protocol when queries are not uniformly distributed.

As to the second security requirement, if two identical queries are posed consecutively without any interfering calls, their calls at the same level will always intersect, hence intersections will give some hint about identical queries. We are also currently studying how to improve the protocol by methodically introducing intersections between non-identical queries to make intersections independent from identical queries.

8 Conclusion

In this paper, we propose a simple, adaptive and deadlock free protocol to hide tree structured data and traversal of it from a data store. Since a lot of data such as XML has a tree structure and queries can be expressed as traversal paths, this protocol can be utilized to hide such data and queries. Compared with existing private information retrieval techniques [4, 9], our protocol does not need replication of databases and it requires less communication, and is thus practical. We provide an example how to apply it to hide XML documents and tree path based queries. Finally, we conduct experiments and observe that the proposed techniques achieve hiding without generating unacceptable concurrency problems.

Acknowledgement

We thank Dr Rida A. Bazzi for his helpful comments on this paper.

References

1. Hacigümüs, H., Iyer, B.R., Li, C., & Mehrotra, S.(2002) Executing SQL over Encrypted Data in the Database-Service-Provider Model, Proceedings of 2002 ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, USA, June 3-6, 2002. pp. 216-227.

2. Oracle Corp.,Database Security in Oracle8i, 1999. Retrieved Febuary 26, 2004, from http://otn.oracle.com/depoly/security/oracle8i/index.html.
3. Smith, S. W., & Safford, D.(2001). Practical Server Privacy with Secure Coprocessors. IBM Systems Journal, Vol. 40, No. 3. pp.683-695.
4. Chor, B., Goldreich, O., Kushilevitz, E., & Sudan, M.(1995). Private Information Retrieval, Proceeding of 36th IEEE Conference on the Foundations of Computer Sciences, Milwaukee, Wisconsin, USA, October 23-25, 1995. pp. 41-50.
5. Bouganim, L., & Pucheral, P.(2002). Chip-secured Data Access: Confidencial Data on Untrusted Servers, Proceedings of 28th Very Large Data Bases Conference, Hongkong, China, 2002. pp.131-142.
6. Bayer, R., & Schkolnich, M.(1977). Concurrency of Operations on B-Trees, *Acta Informatica*, Vol. 9, pp. 1-21.
7. Mohan, C.(1996). Concurrency Control and Recovery Methods for B+-Tree Indexes: ARIES/KVL and ARIES/IM, In Kumar, V.(Ed.) Performance of Concurrency Control Mechanisms in Centralized Database Systems, Prentice-Hall 1996, pp. 248-306.
8. Mohan, C.(2002). An Efficient Method for Performing Record Deletions and Updates Using Index Scans, Proceedings of 28th Very Large Data Bases Conference, Hongkong, China, 2002.pp.940-949.
9. Chor, B., Gilboa, N., & Naor, M.(1997). Private Information Retrieval by Keywords, Technical Report TR CS0917. Technion Israel, 1997.

ANALYSIS OF WEP PERFORMANCE ON MOBILE DEVICES

Arnulfo Ochoa Indart [1], Jesús Arturo Pérez Díaz [2],

1 Informatic Graduate Program, ITESM Campus Cuernavaca, Paseo de la Reforma 182-A,
Cuernavaca, México
Email: 00379337@mor.itesm.mx
2 Electronic and Communications Department, ITESM Campus Cuernavaca, Paseo de la
Reforma 182-A, Cuernavaca, México
Email: jesus.arturo.perez@itesm.mx

Keywords. Wireless Networks, 802.11b, WEP, Performance, Mobile Devices

Abstract. Mobile devices are becoming more popular every day; they must keep up with security implemented by desktop computers. This paper tries to evaluate performance of data transmission with and without ciphering techniques. WEP is not the best way of securing a network but it is widely used, that is why we used WEP on these tests. This article tries to define how much performance is lost with WEP, so we can estimate the loss of performance on mobile devices when TKIP and WPA's MIC protocols are implemented. We observed in the results that decrease on performance was more noticeable on PDAs than other devices such as laptops

1 Introduction

Ever since wireless networks appeared, many questions concerning security issues were made. WEP (Wired Equivalent Privacy) was part of IEEE's 802.11 standard, and it attempted to provide secure wireless communications.

In 802.11 WEP uses a secret 40 bit key (weak) or 128 bit key (strong) in 802.11b and a pseudorandom number generator (RC4). Two processes are applied to clear text; one of them ciphers data and the other one protects it from unauthorized modifications while in transit. The secret key is concatenated with a random initialization vector (IV) that adds 24 bits to the resulting key. This key is processed in the pseudorandom number generator that outputs a large pseudorandom key stream. The transmitter combines it with the clear text using an XOR operation, creates the ciphered text and sends it to the receiver along with the IV. When the receiver gets the ciphered text, it uses the IV and its own copy of the secret key to generate the same key stream as the transmitter. The receiver combines them with the XOR operation and generates the original clear text.

In order to protect the ciphered text against modifications while it is in transit, WEP applies an integrity checking algorithm (CRC-32) to the clear text and generates an integrity check value (ICV).

The ICV is concatenated to the text before it is encrypted with the key and is sent to the receptor along with the IV. When the checking algorithm is applied to the clear text and is compared with the output with the ICV value received, it can be verified if there has been any modification. [1]

However as Nikita Borisov et. al demonstrated, the WEP checksum is a linear function of the message. One consequence of the above property is that it becomes possible to make controlled modifications to a ciphertext without disrupting the checksum. [2].

Description	Processor	RAM	WLAN NIC	OS
Laptop Client 1 – HP ze5785 us	Intel Pentium 4 2.4 Ghz.	512 MB	LAN-Express IEEE 802.11b NIC	Windows XP Home Edition
Laptop Client 2 – IBM Think Pad 2655	Intel Pentium 3 1 Ghz.	128 MB	Proxim IEEE 802.11 b/g PC Card.	Windows 2000 Professional
PDA Client – HP iPAQ 4155	Intel XScale 400 Mhz.	64 MB	Embedded	Windows Mobile 2003
Server Laptop – HP ze5385 us	Intel Pentium 4 2.66 Ghz.	512 MB	LAN-Express IEEE 802.11b NIC	Windows XP Home Edition

WEP uses the RC4 symmetric stream cipher for encryption and decryption purposes. Symmetric means that the sender and receiver must use the same key for proper encryption and decryption functions. [3]

There are other key lengths for WEP, such as 64 bits, which was used in our tests.

There are various types of known attacks against WEP, and it is not considered secure. Although there are other ciphering techniques, WEP is implemented natively in many OS such as Windows XP, Windows Mobile and Palm OS. This is why WEP is still widely used.

Design of secure protocols is difficult, and fraught with many complications. It requires special expertise beyond that acquired in engineering network protocols. A good understanding of cryptographic primitives and their properties is critical. From a purely engineering perspective, the use of CRC-32 and RC4 can be justified by their speed and ease of implementation. [2]
Mobile devices such as PDA’s are being increasingly used in Wireless LANs (WLANs); these devices have limited processing resources; and therefore, the impact on data transfer performance is of particular interest because of the processing overhead it causes.

There are other security protocols such as PEAP or LEAP, which promise better protection, however, it has been proofed that there are other attacks that could affect them such as the ones published by Mishra and Arbaugh, which explains that 802.11 frames, including 802.1X messages, are easily sniffed. For this reason, IEEE 802.11 Task Group I recommends EAP methods resistant to dictionary attack.

It's worth heeding this advice, since dictionary attacks enable an attacker to recover the user password, which often can provide access to more than just the 802.11 network. Therefore these attacks are more serious than the previously documented WEP attacks and customers using 802.1X should strongly consider adopting dictionary attack-resistant authentication methods such as EAP TLS, SRP, TTLS and PEAP. [4]

LEAP is a type of Radius EAP. It is used to authenticate access by a wireless client (typically a laptop or pc) to a wireless router, typically a Cisco Aironet base station.[5]

RADIUS is a widely deployed protocol for network access authentication, authorization and accounting (AAA). [6]

This paper presents an analysis of the data transfer performance achieved by laptops and PDA's when using 64 and 128 bit keys with WEP and when transmitting clear text using an infrastructure WLAN.

2 Experimental Section

2.1 Equipment Used

Two laptops and a PDA were used as clients. A third laptop was used as server. A brief description of the equipment can be found in table 1.

The access point that was used was a Microsoft Broadband Networking Wireless Base Station Model MN-500, which is Wi-Fi certified.

2.2 Performance measurement

In order to obtain performance measurements of common uses of a WLAN, a simple web-based script was written in PHP, running on an Apache 2.0.48 web server with PHP Engine 4.0.1. Measurements were stored using mySQL 4.0.13.

The PHP script sends a random stream of bytes, ranging from 100 to 5000 kilobytes. Three fields are stored in the database, the client's IP address, the amount of data transferred and the time that the transfer took.

The resulting web page is reloaded 5 seconds after the transfer is finished and a new stream of different size is sent to the client.

2.3 Test scenarios

Several tests were performed, in order to test different situations and compare them.

The first variable is the length of the key, three different scenarios were tested in this case, with no key (no WEP encryption), 64 bit, and 128 bit keys.

The second variable is distance, 3 different distances were tested. In every case, all the devices were at the same distance.

a) Five feet away from the Access Point. No interferences.
b) Twelve feet away from the Access Point. No interferences.
c) Forty feet away from the Access Point. On the second floor, home environment (Computers and PDA were on the first floor).

For each scenario, 1200 samples were gathered, 400 for every mobile device.

Using the gathered data, simple statistical analysis was calculated, specifically, the mean value of the samples and the standard deviation.

3 Results and Discussion

3.1 Performance with no WEP encryption

	5ft	12ft	40ft
HP Laptop	170.21	148.9	122.83
IBM Laptop	169.26	145.8	120.46
iPAQ PDA	168.6	148.39	119.14

Table 2: Average results in KB/S with no WEP encryption.

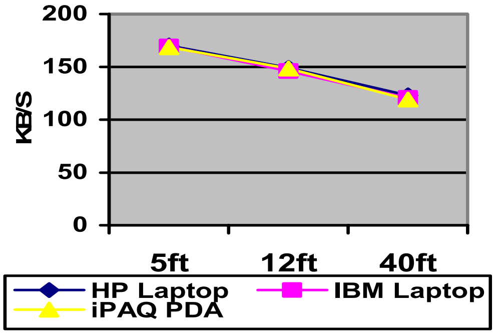


Figure 1: Data transfer performance no WEP encryption

When using no WEP encryption, the performance loss is similar on both laptops and the PDA as shown in Table 2 and Figure 1.

3.2 Performance with 64 bit key WEP encryption

	5ft	12ft	40ft
HP Laptop	162.19	147.32	112.14
IBM Laptop	158.22	149.41	115.11
iPAQ PDA	154.63	141.32	104.78

Table 3: Average results in KB/S with WEP and a 64 bit key.

Test results with a 64 bit key show that the PDA's performance was more noticeable than both laptops. This can be observed in Table 3 and Figure 2.

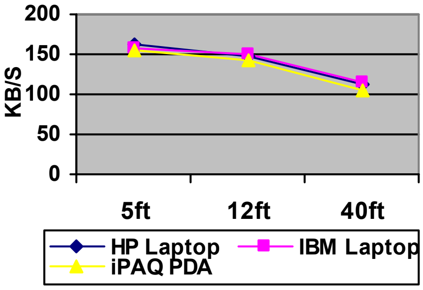


Figure 2: Data transfer performance 64 bit key WEP encryption

3.3 Performance with 128 bit key WEP encryption

	5ft	12ft	40ft
HP Laptop	147.24	140.33	118.75
IBM Laptop	150.81	145.28	117.63
iPAQ PDA	140.29	134.5	90.69

Table 3: Average results in KB/S with WEP and a 128 bit key.

It is clear that the PDA decreased its performance more than laptops. This can be seen in Table 3 and figure 4.

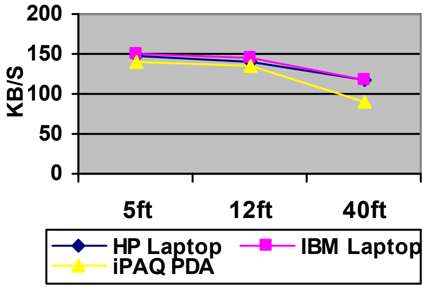


Figure 3: Data transfer performance 128 bit key WEP encryption

3.4 Results Analysis

	No WEP	64 bit	128 bit
HP Laptop	147.31	140.55	135.44
IBM Laptop	145.17	140.91	137.90
iPAQ PDA	145.37	133.57	121.82

Table 4: Overall Performance in KB/S

It is clearly visible that the PDA's performance (See Table 4) was considerably reduced by WEP encryption. It is clear that the reduced computing power of the PDA resulted in a bigger impact on performance.

As mentioned above, WEP uses symmetric keys, because of that, we expected better performance results on the PDA, but it affected it visibly. We would now expect that using EAP-TLS or other similar technique the performance loss to be greater.

TKIP changes the ciphering key very often, and requires much more resources. Based on this, we can extrapolate the results and consider that when using TKIP, the performance loss will be much bigger.

Both laptops had similar behavior, and they were not visibly affected by WEP encryption.

We can see an overall comparison of performance in figure 4.

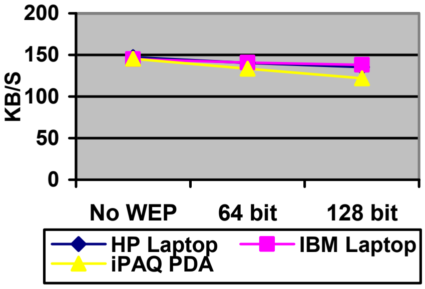


Figure 4: Overall performance comparison

4 Future Works

We will repeat these tests with ciphering techniques specified by WPA and evaluate their performance in order to search alternatives for mobile devices if there is a considerable loss of performance.

5 Conclusions

Approximately, the PDA lowered its performance to 83.80% compared to the 91.94% observed in Client 1 and 94.99 % of Client 2, when looking their performance based on no WEP encryption and 128 bit encryption.

From the standard deviations observed, the PDA had the lowest levels overall, this can be because laptops usually run other processes on the background that might impact some measurements.

Security is vital to wireless communications, there has been a big amount of effort and research to provide reliable ciphering techniques. Progress has been achieved in this field; however there are new scenarios where wireless communications were not very popular a few years ago.
Mobile devices have limited resources and processing power, this is why, ciphering techniques used in these devices, have to meet their constraints and yet meet security levels.

It will be vital to take these constraints when designing new security schemes, and when these schemes are deployed to new operating systems for mobile devices, they must allow limited devices to work properly, without degrading QoS and providing secure, reliable data transfers.

WPA security protocols are expected to consume more resources than old protocols such as WEP, so special protocols for limited devices should be developed, so their performance is not affected.

References

[1] Nichols, Randall and Lekkas, Panos. Wireless Security: Models, Threats, and Solutions. McGraw Hill. Edition 1, 2002. ISBN: 0071380388

[2] Nikita Borisov, Ian Goldberg, David Wagner. Intercepting Mobile Communications: The Insecurity of 802.11 http://www.isaac.cs.berkeley.edu/isaac/wep-draft.pdf

[3] Shon Harris, Latest trends in wireless security http://a907.g.akamai.net/7/907/3644/v0001/ntschool1.download.akamai.com/3644/_vl/Articles/WIRELESS-SECURITY-SHONa.pdf

[4] Arunesh Mishra and William A. Arbaugh, An Initial Security Analysis of the IEEE 802.1X Standard, University of Maryland. http://www.cs.umd.edu/~waa/1x.pdf

[5] Cameron MacNally, Cisco LEAP protocol description http://www.missl.cs.umd.edu/wireless/ethereal/leap.txt

[6] Bernard Aboba. Wireless LAN Security Site. http://www.drizzle.com/~aboba/IEEE/

Risk Based Security Analysis of Permissions in RBAC

Nimal Nissanke and Etienne J. Khayat

Centre for Applied Formal Methods, London South Bank University
103 Borough Road, London SE1 0AA, UK
{Nissanke,E.Khayat}@lsbu.ac.uk

Abstract. Because of its vulnerability to errors and, hence, unauthorised access, assignment of access rights is a critically important aspect of RBAC. Despite major advances in addressing this clearly using formal models, there is still a need for a more robust formulation, especially incorporating strict guidelines on assignment of access rights and how to perform such tasks as delegation of access rights. In this respect, this paper proposes a precise mathematical framework, capable of considering important factors such as the relative security risks posed by different access operations when performed by different users. This is based on a novel concept of a security risk ordering relation on such tasks, to be established by a detailed independent risk assessment process. In the case of lack of information on security risks, the approach makes conservative assumptions, thus forcing the security analyst to re-assess such situations if he disagrees with this default interpretation. The risk ordering relation is central to a security-orientated definition of role hierarchies and a security-risk minimising strategy to role delegation.

1 Introduction

Role Based Access Control (RBAC) is a widely used access control mechanism whereby access rights to users (subjects) are granted on the basis of their roles in an institution rather than as individuals. Allocation of access rights, whether it takes place as a result of system administrator's duties, such as permission assignment to roles, or discretionary actions exercised by subjects higher up in role hierarchy, such as delegation, is a process vulnerable to errors and existence of unforeseen loopholes that could compromise the system security.

The growing interest in RBAC is evident from the large number of works devoted to it. Notable among them are the works [5, 6, 9, 10] characterising a hierarchy of RBAC models with increasing sophistication, dealing with role hierarchies, potential conflict of interests between roles, etc. A major outcome of these developments is the recognition by the research community of the need for a standard [1] aimed at a unified model for RBAC. Related is also our own work [7], providing a formal state-based model for the core RBAC [1]. Important issues related to delegation are elaborated in a number of works; with [2] dealing with a basic but sufficiently detailed model of delegation, [8, 11, 12] showing a practical scenario of the implementation of delegation and [13] giving detailed mathematical models for various types of delegation.

Despite the above advances, there seems to be little definitive rules or guidelines that govern the assignment of access rights in RBAC and clarify the means by which the goals of the security mechanisms are to be achieved. System invariants for access rights allocation that should always be respected are not sufficiently well defined in the literature, including those cited above. Allocation of access rights is often based on informal rules based on past experience or inherited institutional practices. In order to overcome the above deficiencies, this paper introduces a novel approach ensuring a consistent and systematic interpretation of security requirements and a compatible and effective way to enforce the security arrangements.

Our approach is based on the concept of a *risk ordering relation* [4] expressing the relative risk posed by a subject of a particular role performing a particular task, compared to the same posed by a similar subject-task combination. It is a mathematical concept designed both to introduce rigor into security modelling and to eliminate ambiguities, omissions and inconsistencies in the risk assessment process. Risk ordering itself is established through an appropriate independent security risk analysis of the organisation. From the perspective of security risk analysis, the approach offers two major benefits: firstly, it makes explicit the form of the information required from such an analysis and, secondly, it prompts the security expert to question his security assessment, thereby improving the quality and comprehensiveness of the process, as well as the end–product of the security risk assessment. Questioning of the security assessment is achieved by a default conservative interpretation of risks levels whenever there is a lack of information on security risks. According to this, any task with a possible inadequate consideration of risk is placed conservatively in a lower *security risk band* by default, alerting the security analyst to reconsider its risk nature if such an interpretation is undesirable. Turning to the modelling of RBAC, our work proposes certain security principles for permission assignment to roles and for subject-invoked role delegation.

The paper has the following structure. Section 2 introduces the required basic concepts of RBAC and the relevant mathematical definitions used later. Section 3 presents the concept of *security risk ordering relation*, expressing the risks posed by different combinations of roles and tasks (permissions) relative to one another. Section 4 states the proposed principles of allocation of access rights in RBAC in a precise manner. Section 5 presents a case study drawn from the domain of health care illustrating the application of the latter principles. Section 6 concludes the paper with a summary of achievements.

2 Basic Concepts and Mathematical Preliminaries

The purpose of RBAC is to determine at run–time whether to allow, or deny, a user (*subject*) accessing a required resource (*object*) based on access rights granted to the roles that subjects perform in the organisation. This section introduces the basic RBAC concepts relevant to these issues and an appropriate notation for the discussion; see also [7]. Our formulation is based on the following basic types of entities: *SUBJECT* denoting the set of all possible users (subjects) of the computer system (including any non-human agents), *OBJECT* the entities (objects) being accessed by the subjects, *ROLE* the roles in the capacity of which the subjects derive the access rights to the objects concerned, and *OPERATION* the set of operations that may be performed on the objects. Disregarding here the applicability of operations to specific objects, the set of all possible tasks are denoted by *TASK*, defined as

$$TASK = OPERATION \times OBJECT \tag{1}$$

Associated with the above are the following functions [Note: $\mathbb{P}$ denotes the power set of its operand set (on the right)]:

$$SubjectRoles : SUBJECT \rightarrow \mathbb{P}\, ROLE \tag{2}$$

$SubjectRoles(s)$ giving the set of roles associated with each subject s, and

$$Permissions : ROLE \rightarrow \mathbb{P}\, TASK \tag{3}$$

$Permissions(r)$ giving the set of tasks authorised for each role r. When dealing with individual permissions, it is convenient to have the elements of the following set

$$PERM \subseteq ROLE \times TASK \tag{4}$$

the elements of which denote roles and the applicable tasks.

In relation to delegation of access rights or tasks in hierarchical RBAC, we introduce two types of delegation: lateral delegation (a role delegating its duties to another role lying at the *same* level of the hierarchy) and downward delegation (a role delegating its duties to a junior role). A record of such delegated roles to each particular role may be maintained by the following functions:

$$lat\_del\_roles, down\_del\_roles, delegated\_roles : ROLE \rightarrow \mathbb{P}\, ROLE \tag{5}$$

$lat\_del\_roles(r)$, $down\_del\_roles(r)$ and $delegated\_roles(r)$ giving, respectively, the roles delegated to role r laterally, downward and in total. Note that for each of the above, r cannot be delegated to itself. Together the above satisfy

$$\forall\, r \in ROLE \bullet delegated\_roles(r) = lat\_del\_roles(r) \cup down\_del\_roles(r) \tag{6}$$

Turning attention to conflicts of interests (COI), there are two kinds of separation of duties that need to be taken into account in determining the permitted delegations of roles and tasks, namely: a) Static Separation of Duty (SSD), which concerns the prevention of any conflict of interests arising from the mere assignment of such roles to the same subject, and, b) Dynamic Separation of Duty (DSD), which concerns the concurrent exercise of such roles by any subject at the same time and not whether they can be assigned to the same subject. With the above in mind, let us introduce three symmetric and irreflexive binary relations *SSD*, *COI* and *COI* on *ROLE*, such that $COI = SSD \cup DSD$.

3 Security Risk Ordering

In general, risk expresses a combined measure of the likelihood of a hazardous, or a harmful, event occurring and the ensuing consequences should it ever take place. In computer security, such events include intrusion, tampering with data, eavesdropping, etc., violating system security properties such as *confidentiality*, *integrity* and *availability*. Security threats not intensifying, the risk of such events taking place usually reduces with increasing protection. Risk assessment is an exercise in its own right and is beyond the scope of this paper. What is important here is, however, the outcome of the risk assessment process and, in particular, the relative risks posed by various security threats relative to one another.

Risk ordering relation, introduced here, relies on a comparison of risks arrived at by an appropriate independent risk assessment process. It is denoted by $\sqsubseteq$ and has the form

$$\sqsubseteq : PERM \leftrightarrow PERM \tag{7}$$

Its meaning is such that, given two permissions p_1 and p_2, where $p_1, p_2 \in PERM$, $p_1 \sqsubseteq p_2$ signifies that p_2 is more, or equally, secure compared to p_1 or, alternatively, p_1 carries a higher, or an equal, security risk compared to p_2. $\sqsubseteq$ is reflexive and transitive, but not necessarily symmetric or antisymmetric. We decompose $\sqsubseteq$ into two relations:

- $\preccurlyeq$: a partial order relation over the elements of *PERM*, which orders their risk levels. If $p_1 \preccurlyeq p_2$, then p_1 carries a higher security risk than p_2, unless p_1 and p_2 denote the same permission.
- $\approx$: an equivalence relation between the elements of *PERM*. If $p_1 \approx p_2$, then p_1 and p_2 are identical in terms of security risk.

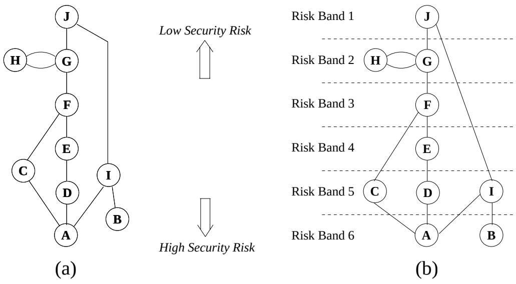
As a consequence of this decomposition, $\sqsubseteq$ is the union of $\preccurlyeq$ and $\approx$. In other words, for permissions p_1 and p_2, $p_1 \sqsubseteq p_2$ if and only if $p_1 \preccurlyeq p_2$ or $p_1 \approx p_2$.

The relation $\sqsubseteq$ is best depicted in the form of a graph, as in Figure 1(a), showing the ordering of the permissions. Since the risk analysis is performed by human security analysts, the relation $\sqsubseteq$ may contain gaps, inaccuracies and inconsistencies. Therefore, following [4], we use the concept of *risk band* to alert the risk analyst to such deficiencies. The idea is to interpret any lack of information conservatively in favour of greater provision of security. In effect, risk bands extend the graph of $\sqsubseteq$ with numerically indexed risk bands such that permissions carrying relatively greater security risks are placed in higher risk bands, while the more secure permissions in lower risk bands; see Figure 1(b). In the event of insufficient information as to where a particular permission is to be placed, it is interpreted as an indication that the permission concerned is to be placed in the highest possible risk band, subject to any constraints imposed by other pairs in the relation $\sqsubseteq$. Any disagreement with this default interpretation obliges the security risk analyst to clarify the relative risk levels of the permissions concerned more accurately, thus helping to refine the risk ordering relation and, thereby, making it more complete, accurate and consistent with the required security requirements. The graph of the relation $\sqsubseteq$, extended with risk bands, is referred to as the *risk graph*; see Figure 1(b). The arcs in the graph are assumed to run upward and the reflexivity of the permissions in the relation $\sqsubseteq$ are not shown in the graph to reduce clutter. The risk bands are numbered from 1 to some *n*, higher indices signifying greater risk. Risk graph, corresponding to a specific relation $\sqsubseteq$, is to be determined according to the following rules:

- Permissions with the highest security risk, or the least secure ones, (i.e. those in the *n*th risk band) are exactly:
  a) The permissions that are lowest in the partial order relation $\preccurlyeq$, but not related by $\approx$ to any other permission in $\preccurlyeq$.

b) Any other permissions related by $\approx$ to the ones just mentioned in (a) above.

- If there exist two distinct permissions p_1 and p_2 such that a) $p_1 \preccurlyeq p_2$, b) p_1 is the only immediate predecessor so related to p_2, and c) p_1 is in risk band i, then p_2 is in risk band $(i-1)$. If p_2 has several immediate predecessor permissions, then its risk band index would be one less than the lowest risk band index of those predecessor permissions.
- If there exist two permissions p_1 and p_2 such that $p_1 \approx p_2$, then p_1 and p_2 are in the same risk band.



(a)

(b)

Fig. 1. (a) Risk ordering relation (b) Risk graph.

Associated with the risk graph is a risk distance between two permissions of the form: $RD(p_1, p_2) = RB(p_1) - RB(p_2)$, where $RB(p)$ gives the risk band index of a given permission $p \in PERM$, taking the sign into consideration. From the security risk perspective, two permissions p_1 and p_2 are said to be *risk-comparable* if and only if they are equivalent through $p_1 \approx p_2$ or are in different risk bands (i.e., $RB(p_1) \neq RB(p_2)$). If they are in the same risk bands (i.e., $RB(p_1) = RB(p_2)$), but are not equivalent (i.e., $p_1 \not\approx p_2$), then they are said to be *risk-non-comparable*.

4 Principles of Allocation and Delegation of Permissions

This section formulates several principles to be followed when allocating access rights. These concern the cases of permission assignment to roles and delegation of access rights.

4.1 Relations on Roles

The hierarchical model of RBAC [1], also known as RBAC$_1$ [10], places the roles in a hierarchy in accordance with the functional requirements of the organisation and other considerations such as the skills, the competence, the past experience, etc., required as part of the job descriptions. However, this is based mathematically on a simple set–theoretic characterisation of roles as a partial order $\leqslant$, namely, for any two roles r_1 and r_2 as

$$r_1 \leqslant r_2 \Rightarrow Permissions(r_1) \subseteq Permissions(r_2) \tag{8}$$

It is important to note that (8) characterises only a hierarchical relationship between roles with inheritance of permissions of juniors by seniors. In our view, however, there are other notions of seniority relations of relevance to security. Of particular interest here is a relation that characterises roles performing different kinds of activities but being equivalent. This is because, for example, being equivalent in status would allow the delegation of roles that deal with authorisations, etc. With this in mind, this work uses three relations on roles, two of them being

- A partial order relation, $\leqslant$, as defined in (8), dealing with hierarchical inheritance of permissions of junior roles by their seniors.
- An equivalence relation, $\simeq$, dealing with equivalence of roles belonging to different categories of roles in terms of their status.

leaving the third relation for Section 4.2. In relation to $\leqslant$ and $\simeq$, as an example, consider the members of a hospital in two different role categories: *medical* and *nursing*. According to $\leqslant$, roles in the *medical* category may be ordered hierarchically as: *resident* $\leqslant$ *surgeon* $\leqslant$ *consultant*, whereas those in the *nursing* category as *nurse* $\leqslant$ *senior_nurse* $\leqslant$ *chief_nurse*. Furthermore, using the equivalence relation $\simeq$, it is possible to relate the chief nurse and the surgeon as *chief_nurse* $\simeq$ *surgeon* in order to convey that they have the same seniority status and, therefore, they are eligible to delegate, for instance, certain authorisation tasks between them.

4.2 Principle I: Permission Assignment to Roles

As noted above, roles in RBAC are assigned permissions by associating them with the tasks that they are authorised to perform. In most cases, this association is based solely on the functional requirements of the organisation. Prior to such assignment of permissions to roles, however, a security risk assessment needs to be performed in order to verify if the functional requirements would induce any unintended security threat to the organisation's assets. This is where the security risk ordering relation, introduced in Section 3, proves to be useful. With further implications in terms of risk bands, the risk graph of $\sqsubseteq$ represents a detailed ordering of security risks posed by different permitted role–task combinations.

The third hierarchical relation on roles, introduced in this work, takes into account the risks described above. It is a hierarchical partial order and is denoted by $\ll$. It extends the relation $\leqslant$ in (8) by incorporating $\sqsubseteq$ and, following [4], is defined as

$$\begin{aligned} r_1 \ll r_2 \Leftrightarrow (r_1 < r_2) \wedge (\forall\, t \in \mathit{TASK} \bullet t \notin \mathit{Permissions}(r_1) \wedge \\ t \in \mathit{Permissions}(r_2) \Rightarrow (r_1, t) \preccurlyeq (r_2, t)) \end{aligned} \tag{9}$$

According to this principle, role r_2 is senior to r_1, i.e., $r_1 \ll r_2$, if and only if the role r_2 is senior to the r_1 in the sense of $\leqslant$ in (8), i.e., $r_1 < r_2$, and all permissions, which are not included in the junior role r_1 but are in r_2, are handled more securely by r_2 than by r_1 with respect to the relevant risk graph. The intention is to ensure that senior roles, while inheriting permissions of the respective junior roles, are entrusted with certain permissions requiring greater degree of security. This is a justification for a security–orientated notion of a hierarchical seniority. However, this does not necessarily mean that the senior role can handle all its permissions more securely than the junior role. In fact, it may be the case that the junior role is intended to handle its own tasks, perhaps with the exception of its own inherited ones, more securely than the senior role because of, for example, the specialist expertise required by the tasks concerned.

4.3 Principle II: Delegation of Tasks

Our approach to delegation of access rights is based on certain rules that take security risks into consideration. The lack of such explicitly stated rules in other works may be due to the informality of the way delegation is handled normally or the excessive number of possibilities in delegation encountered in practical situations. Note that delegation applies only to *level 1 delegation* [3], that is, to roles initially assigned by the system administrator and not to those gained by previous delegations from other roles.

Principle II(a): Lateral Delegation of Tasks. The lateral delegation here concerns the delegation of roles at the same level of seniority as understood by the relation $\simeq$, introduced in Section 4.1. This may be expressed as

$$\forall r_1, r_2 \in ROLE \bullet r_1 \neq r_2 \wedge r_2 \in lat\_del\_roles(r_1) \Rightarrow r_1 \simeq r_2 \tag{10}$$

Principle II(b): Downward Delegation of Tasks. This principle deals with the delegation of its access rights by one role to another in a strictly lower level in the hierarchy $\ll$; see Section 4.2. Let us deal here only with the total delegation, i.e., the delegation of all access rights of the delegator role [3]. In order for such a delegation to be permitted, the two conditions (11) and (13) are to be satisfied.

Firstly, the delegating and delegatee roles must be hierarchically related, as in

$$\forall r_1, r_2 \in ROLE \bullet r_1 \neq r_2 \wedge r_2 \in down\_del\_roles(r_1) \Rightarrow r_1 \ll r_2 \tag{11}$$

Secondly, security risk considerations need to be taken into account. To minimise security risks, the access rights are better be delegated to the role(s) that would present the least risk when they perform the delegated tasks. This can be established using the risk graph, introduced in Section 3. Considering each of the tasks to be delegated under the delegating role, it is possible to calculate the worst (lowest, taking the sign into account) risk distance from the delegating role to each candidate delegatee role. Thus, for each potential pair of delegating-delegatee roles there is to be a lowest risk distance. The delegatee role giving the largest of these risk distances (taking the sign into account) would be the one to be favoured for delegation. With this in mind, let us first define the worst risk distance between the permissions of one role r_1 relative to the same permissions under another role r_2

$$\begin{gathered}\forall r_1, r_2 \bullet r_2 \ll r_1 \Rightarrow worst\_risk\_dist(r_1, r_2) = \\ min\{RD((r_1, t), (r_2, t)) \mid t \in permissions(r_1) \wedge t \notin permissions(r_2)\}\end{gathered} \tag{12}$$

where *min S* gives the minimum value in the set *S* (of integers). The role(s), which is the least risky for delegating r_1's permissions, is a role r_2 having the largest among the worst risk distances calculated as described above. In other words, for delegating r_1 the least risky delegatee role is r_2, provided that

$$\forall r_3 \bullet r_3 \ll r_1 \Rightarrow worst\_risk\_dist(r_1, r_2) \geqslant worst\_risk\_dist(r_1, r_3) \tag{13}$$

Principle II(c): Avoidance of Conflicts of Interest. Furthermore, neither of the above forms of delegation should result in any static conflict of interest with other delegated roles and the target (delegatee) role (*r* below) itself. That is, delegation must respect the static separation of duty. This principle may be expressed as

$$\begin{aligned} &\forall\, r \in \mathit{ROLE} \bullet \exists\, \mathit{roles} \in \mathbb{P}\,\mathit{ROLE} \bullet \mathit{roles} = \mathit{delegated\_roles}(r) \cup \{r\} \Rightarrow \\ &\qquad \mathit{roles} \times \mathit{roles} \cap \mathit{SSD} = \emptyset \end{aligned} \tag{14}$$

5 Case Study: A Health Care Information System

This section illustrates the proposed approach using a hypothetical, but realistic, simple access control system applicable to a hospital environment, but in relation to: a) the construction of a role hierarchy (Principle I), and b) downward delegation (Principle IIb), both based on security considerations. A description of the functional requirements of the access control system are summarised, along with the notation, in Table 1.

Table 1. The tasks defined in the hospital's information system

Task Name	Representation	Brief Description	Authorised Roles[†]
t_1	(lead,op)	leading an operation	consultant (c)
t_2	(asst,op)	assist in performing an operation	consultant (c) surgeon (s)
t_3	(prep,pat)	pre-operation care for a patient	nurse (n)
t_4	(mont,pat)	post operation monitoring of patient	nurse (n)
t_5	(adm-med,pat)	administering medication to patient	nurse (n)
t_6	(adm-aneas,pat)	administering anaesthetics to patient	anaesthetist (a).

[†] Note: Shown in brackets is the notation to be used later.

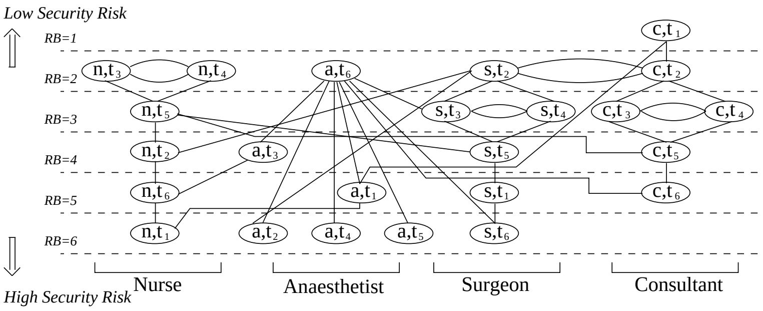


Fig. 2. Risk graph of the permissions in the hospital's information system.

A security risk assessment, involving the permissions and the roles concerned, has resulted in a security risk ordering relation shown in Figure 2. Though its primary purpose is to present relative security risk levels between various pairs of permissions, belonging also to different roles, it also indicates the risk graphs of individual roles. Arcs on the graphs, assumed to run upwards, show the risk–comparability between permissions.

Let us first consider three possible role hierarchies in relation to both the Principle I and the functional requirements. These are shown in Figure 3 along with the permissions associated with each role. The three role hierarchies can be checked for conformity with (9) against the security risk graph shown in Figure 2. By the manner of their construction, all three hierarchies satisfy the relation $\leqslant$ in (8) – the first conjunct of (9). Hierarchy 1 satisfies also the second conjunct. In this case, note that $\mathit{Permissions}(s) \subset \mathit{Permissions}(c)$, $t_1 \in \mathit{Permissions}(c)$ but $t_1 \notin \mathit{Permissions}(s)$ and, according to the risk graph, $(s, t_1) \preccurlyeq (c, t_1)$. Analogous arguments apply to pairs of roles *s* and *n*, and *a* and *n*. It may be noted that Hierarchy 1 also satisfies the functional requirements. Following a similar analysis, we note that Hierarchy 3 conforms with Principle I, but violates the functional requirements. In Hierarchy 2, however, in relation to the pair $n \leqslant s$ (by transitivity of $\leqslant$), $t_6 \in \mathit{Permissions}(s)$ and $t_6 \notin \mathit{Permissions}(n)$ but $(s, t_6) \preccurlyeq (n, t_6)$, which violates Principle I. Thus, we conclude that only Hierarchy 1 satisfies both the functional requirements and the security considerations expressed in Principle I, thus justifying its applicability to RBAC as proposed here.

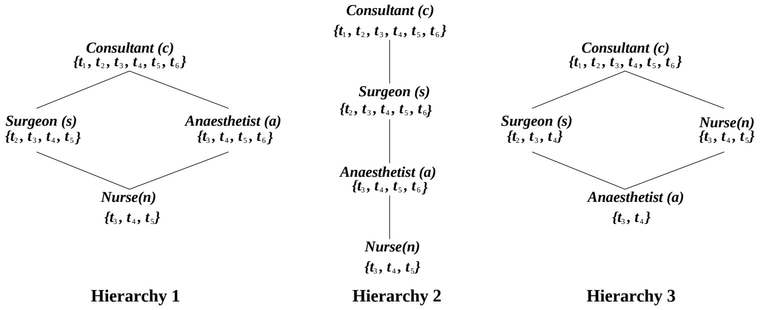


Fig. 3. Three possible role hierarchies

Turning attention to delegation, let us consider a situation where a subject exercising the role *consultant* wishes to delegate his role (i.e. the totality of the tasks) to a junior role in Hierarchy 1. In determining the most secure role(s) to whom the delegation should take place, the risk distances between the permissions of *consultant* and the same performed by the other roles need to be calculated. Note, however, that some of *consultant*'s tasks are shared also by the junior roles. Therefore, risk distances are needed only in relation to the non–shared tasks. The tasks concerned are: t_1 and t_6 in the case of delegation to *surgeon*; t_1 and t_2, in the case of *anaesthetist*; and t_1, t_2 and t_6 in the case of *nurse*. Therefore, according to (12), the worst risk distances are, respectively, -4, -4 and -5, leading to the least risky roles *anaesthetist* and *surgeon* for delegating *consultant*'s role.

6 Conclusion

This paper presents a rigorous formal approach for dealing with some of the key issues in RBAC, in particular, delegation and allocation of access rights. Assignment of access rights is a critical and an error-prone process. Therefore, precise, clear and well-studied guidelines are essential for combating security breaches resulting from unauthorised access rights. An important contribution of the proposed approach, in this respect, is the formulation of several principles for defining role hierarchies and handling role delegation based on a novel idea of a security risk ordering relation. The approach also incorporates precise ways to consider other factors, such as functional requirements and conflicts of interest, etc., essential for assuring the system integrity. The risk ordering relation relies on a detailed assessment of the risks faced by the system. In the event of lack of sufficient information, the approach enforces certain default interpretations of risk in a conservative manner, so that any disagreement leads to a refinement of the security risk analysis. A case study drawn from health care domain illustrates the approach and demonstrates its effectiveness.

References

1. American National Standard for Information Technology. *Role Based Access Control.* Draft BSR INCITS 359, April 2003.
2. Barka E. and Sandhu R. *A Role-Based Delegation Model and Some Extensions.* Proceedings of the 23rd NIST-NCSC National Information Systems Security Conference, pp: 101–114, Baltimore, USA, October, 2000.
3. Barka E. and Sandhu R. *Framework for Role-Based Delegation Models.* Proceedings of the 16th IEEE Annual Computer Security Applications Conference, pp: 168–175, New Orleans, Louisiana, USA, December, 2000.
4. Dammag H. and Nissanke N. *A Mathematical Framework for Safecharts.* Proceedings of the 5th International Conference of Formal Engineering Methods, pp: 620–640, Singapore, Singapore, November, 2003.
5. Ferraiolo D. Cugini J., and Kuhn R. *Role-Based Access Control (RBAC): Features and Motivations.* Proceedings of the 11th Annual Computer Security Applications Conference, pp: 241–248, New Orleans, LA, USA, December, 1995.
6. Ferraiolo D., Sandhu R., Gavrila S., Kuhn R. and Chandramouli R. “Proposed NIST Standard for Role-Based Access Control”. *ACM Transactions on Information and System Security (TISSEC)*, Vol. 4, No. 3, August 2001, pp: 224–474.
7. Khayat E. and Abdallah A. *A Formal Model for Flat Role-Based Access Control.* Proceedings of the ACS/IEEE Conference on Computer Systems Applications, Tunis, Tunisia, July, 2003.
8. Na S. and Cheon S. *Role Delegation in Role-Based Access Control.* Proceedings of the 5th ACM workshop on Role-Based Access Control, pp: 39–44, Berlin, Germany, June, 2000.
9. Sandhu R., Coyne E., Feinstein H. and Youman C. “Role-Based Access Control Models”. *IEEE Computer*, Vol. 29, No. 2, November 1996, pp: 38–47.
10. Sandhu R., Ferraiolo D. and Kuhn R. *The NIST Model for Role-Based Access Control: Towards A Unified Standard.* Proceedings of 5th ACM Workshop on Role-Based Access Control, pp: 47–64, Berlin, Germany, July, 2000.
11. Zhang L., Ahn. G.J. and Chu B.T. “A Rule-Based Framework for Role-Based Delegation and Revocation”. *ACM Transactions on Information and System Security*, Vol. 6, No. 3, August 2003, pp: 404–441.
12. Zhang L., Ahn. G.J. and Chu B.T. *A Role-Based Delegation Framework for Healthcare Information Systems.* Proceedings of the 7th ACM symposium on Access Control Models and Technologies, pp: 125–134, Monterey, California, USA, June, 2003.
13. Zhang X., Oh S. and Sandhu R. *PBDM: A Flexible Delegation Model in RBAC.* Proceedings of the 8th ACM symposium on Access Control Models and Technologies, pp: 149–157, Como, Italy, June, 2003.

Towards a Classification of Security Metrics ⋆

Carlos Villarrubia, Eduardo Fernández-Medina, and Mario Piattini

Universidad de Castilla - La Mancha, Alarcos Research Group
Paseo de la Universidad, 4, 13071, Ciudad Real(Spain),
{Carlos.Villarrubia, Eduardo.FdezMedina, Mario.Piattini}@uclm.es

Abstract. For the generation of trust in the use of information and communications technologies it is necessary to demonstrate security in the use of these technologies. Security metrics or assurance metrics are the most appropriate method to generate that trust. In this article we propose a series of features for classifying security metrics. We present the main conclusions obtained through this classification together with the list of metrics analyzed.

1 INTRODUCTION

The information and support processes, systems and networks are important assets to any organization. These assets suffer risks and insecurities continually coming from a wide variety of sources, including threats based in malicious code, programming errors, carelessness of people, sabotages or fires.

According to [1], the loss due to malicious code alone exceeded $13 billion in 2001, and security expenditures are projected at more than $3 billion in 2004.

This concern has prompted many organizations and investigators to propose different metrics to evaluate the security of their information system. In general, there exists a consensus in affirming that the election of the metric depends on those concrete security necessities of each organization. Most of the analyzed proposals propose methodologies for the election of these metrics [2–7]. Even in some cases the necessity is suggested of developing specific methodologies for each organization [8].

In any one of these proposals the necessity is to quantify the different relative aspects of security to be able to understand, to control and to improve the trust in the information system.

If an organization doesn't use security metrics to make decisions, the choices will be motivated by purely subjective aspects, external pressures or even purely commercial motivations.

⋆ This research is part of the CALIPO project, supported by the Dirección General de Investigación of the Ministerio de Ciencia y Tecnología (TIC2003-07804-C05-03), and the MESSENGER project, supported by the Consejería de Ciencia y Tecnología of the Junta de Comunidades de Castilla-La Mancha (PCC-03-003-1).

2 SECURITY METRICS

2.1 Metrics Classification

To analyze the different metric proposals it is necessary to use certain approaches to classify them and to be able to obtain conclusions.

The selection of these classification approaches is based on the different previous proposals [9, 3, 4, 7], keeping in mind that they cover the different necessities of the security of an organization, eliminating the repetitions of proposed approaches and selecting those approaches with greater generality.

The approaches selected to classify the security metrics correspond to the different objectives of security pursued, to the control area used to get those objectives, the moment that those controls are applied and to the audience directed with that metric.

1. Security Objective (SO). The security of a system is characterized by information like the persecution of the following objectives:
   - Confidentiality, assuring that only those who are authorized can access the information.
   - Integrity, assuring against the unauthorized modification of the information.
   - Availability, assuring that the authorized users have access to the information and their assets associates when they require it.
   - Authentication, assuring that the identity of a subject or resource is the one claimed.

   In our study, we have included a general objective to characterize those metrics that pursue two or more objectives of security.
2. Control Area (CA). The previous objectives are achieved using different controls in the information system. According to [9], those different types of controls to get the objectives of security can be classified as:
   - Management. Security controls that can be characterized as managerial. In general, they focus on the management of the computer security program and the management of the risk within the organization.
   - Operational. Security controls implemented and executed by people (as opposed to systems).
   - Technical. Security controls that the computer system executes.
3. Temporal Dimension (TD). From the point of view of the risks management, the used controls can be applied in different instants:
   - Preventive. Designed to lower the amount and impact of threat.
   - Detective. Used to detect threat once it has occurred.
   - Corrective. Implemented to help mitigate the impact of a loss event.
   - Recovery. They allow the recovery of the system to the state previous to the attack.
4. Intended Audience (IA). The security metrics are the fundamental mission to inform on the different aspects of security. [7] classifies a metric depending on the following intended audience:
   - Technical. Technical personnel of the company or institution.
   - Decision Makers. Different people responsible for the company.
   - External Authorities. Any external entity to the company that should inform on the situation of the security of the company.

2.2 Metrics features

The information of the previous paragraph can be even more valuable to the stakeholders if it comes accompanied by additional information on the metrics themselves, which may help discriminate between metrics with the same functionality and purpose. Based on the proposal of [10], we will distinguish six features for a given metric. The first group identifies three of the basic (intrinsic) properties of any metric. The three remaining features determine whether the metric has been validated or not, the kind of validation used (theoretical or empirical), and whether the metric has a tool that automates its measurement process or not.

1. Objectivity/Subjectivity (O/S). A metric is objective if its values are calculated by an algorithm or a mathematical formula. On the contrary, a metric is subjective if its measurements are (totally or partially) provided by a human being. In case of subjective metrics, it is very important to record the person or expert that performs the evaluation and provides the values.
2. Direct/Indirect (D/I). According to ISO 9126, a direct measure is a measure of an attribute that does not depend upon a measure of any other attribute. An indirect measure is a measure of an attribute that is derived from measures of one or more other attributes.
3. Run-time/Static (R/S). This characteristic classifies a metric depending on the moment in which it can be measured. Run-time metrics can only be measured during system operation, acting on instances of the component or system being evaluated. Static measures can be evaluated based on the component properties only. Examples of run-time measured metrics are percentage of used media sanitized before reuse or disposal and number of intrusion attempts reported. Static measured metrics include percentage of systems that have a contingency plan or percentage of laptops with encryption capability for sensitive files.
4. Theoretical Validation (TV). The main goal of theoretical validation is to prove that a metric actually measures what it is supposed to measure [11]. Theoretical validation can also help us know when and how to apply the metric. This feature indicates whether the metric has been theoretically validated or not, and how. Even though several methods and principles have been proposed for metric theoretical validation (mainly in the context of software engineering), there is no widely accepted proposal yet. The two major approaches currently proposed are the following:
   - Measurement-theory based approaches such as those proposed by [12], [13], and [14].
   - Property-based approaches (also called axiomatic approaches), such as those proposed by [15] and [16, 17].
5. Empirical Validation (EV). Empirical validation tries to demonstrate with real evidence that the metrics meet their objective, and that they are useful in practice. There are three major types of empirical research strategy:
   - Experiments. Experiments are formal, rigorous and controlled investigations. They are launched when we want control over the situation and want to manipulate behavior directly, precisely and systematically. Hence, the objective is to manipulate one or more variables and control all other variables at fixed

levels. An experiment can be carried out in an off-line situation, for example in a laboratory under controlled conditions, where the events are organized to simulate their appearance in the real world. Experiments may alternatively be carried out on-line, which means that the research is executed in the field under normal conditions [18, 19].

   - Case Studies. A case study is an observational study, i.e., it is carried out by the observation of an on-going project or activity. The case study is normally aimed at tracking a specific attribute or establishing relationships between different attributes. The level of control is lower in a case study than in an experiment [20].
   - Surveys. A survey is often an investigation performed in retrospect, when, for example, a tool or technique has been in use for a while. The primary means of gathering qualitative or quantitative data are interviews or questionnaires. These are completed by taking samples which are representative of the population to be studied. The results from the survey are then analyzed to derive descriptive or explanatory conclusions. Surveys provide no control over the execution or the measurement, though it is possible to compare them to similar ones [21].

6. Automation (A). This feature indicates whether the metric has specific tool support or not. Not only methodological, but also technological support is definitely required for the effective use of metrics in industrial settings [22].

3 ANALYSIS OF EXISTING SECURITY METRICS

As mentioned in the introduction, we are currently witnessing a proliferation of metrics for security. For the present study, we surveyed the existing literature on these topics, looking for metrics that could provide interesting information for description, comparison or prediction of any aspect related to the security of an information system. Interestingly, we had to discard some of the metrics because they didn't have a sufficient description to be able to determine the values of those characteristics used to classify these metrics. Examples of these metrics include those used as examples in those articles that describe methodologies for the construction of these metrics. We also discarded repeated metrics, i.e., those metrics proposed by more than one author. We included one instance of such metrics only. Finally, 57 metrics from 85 different proposals were selected, which are listed in the Appendix of this paper.

Regarding the specific classification approaches to security, the results have been the following:

- Security Objective: 74% of the metrics were general, while 9% of the metrics were to do with availability and authentication.
  7% were confidentiality metrics and only one was specific to the integrity.
- Control Area: 44% of the metrics were operational, 30% were relative of technical, and the rest were management.
- Temporal Dimension: 84% of the metrics were preventive metrics, 9% were detective metrics, and 2% were corrective metrics and recovery metrics respectively.

- Intended Audience: 44% of the metrics were for decision makers, 39% were for technical people and the rest for external authorities.

After evaluating the features of the metrics, the following list shows a summary of the results obtained.

- Objectivity or Subjectivity: 96% of the metrics were objective, the rest subjective.
- Direct or Indirect: 61% of the metrics were indirect, the rest were direct.
- Static or Run-time: 63% of the metrics were static metrics, the rest were run-time.
- Theoretical validation: None of the surveyed metrics had been theoretically validated.
- Empirical validation: Only one of the metrics had been empirically validated- even worse, and none of the rest of the proposals mentioned empirical validation as something they were planning to achieve as part of their future work.
- Automation: Only one of the metrics had some kind of supporting tool.

These results provide a global picture the profile of the surveyed metrics:

- As expected, most of the metrics defined are general metrics and this type of metric only measures general actions relative to the security and in an indirect way they have the preservation of the confidentiality, integrity and availability as objectives.
- Most of the metrics are of a preventive character showing the importance granted to avoidance of problems of security.
- Regarding the area of the used controls and intended audience they have there exists a reasonable balance indicating that the metric proposals form correct aspects.
- Most of the metrics are objective. This is good, since this kind of metrics are more reliable and easier to automate.
- Most of the metrics are direct metrics. Although these metrics are very important, they are only a first step towards the final goal of satisfying the information needs of a user. Hence, indirect metrics, which usually provide more information than direct metrics, and indicators based on them should also be defined
- The lack of validation and automation of the metrics is common to all the disciplines in which the application of metrics is still immature, and clearly shows an area of research that needs to be addressed in order to be able to rely on real engineering methods and tools.

4 CONCLUSIONS AND FUTURE WORK

In this paper we have presented the results of a survey we have conducted on the most representative existing security metrics.

The results obtained show the distribution of the metrics and, more importantly, the areas with lack of metrics which therefore require the definition of new metrics, specific for these areas.

There are several possible extensions to our work. In the first place, we need to continue classifying forthcoming metrics, in order to confirm and validate the conclusions extracted from this first classification, and to further analyze the tendencies in the time for the proposition of new metrics.

We also want to start analyzing the relative importance among those metrics for the attainment of the objectives of security. In this way, additional approaches will be used to prioritize the use of metrics. We also want to analyze the difficulty in the obtaining of the metric ones or in their use to guide in the modification of those metrics to be more useful.

The characterization of security metrics proposed is not complete because some metrics are the same values for all features. A future work is to refine this characterization so that each metric is different in the classification.

Finally, indicators should be defined in function of the size of the organization and sector (for example, public sector and private sector) because it is not realistic to have a good group of metrics which are useful for all the organizations.

References

1. Mercuri, R.T.: Analyzing security costs. Communications of the ACM **46** (2003) 15–18
2. Swanson, M., Bartol, N., Sabato, J., Hash, .J., Graffo, L.: Security metrics guide for information technology systems. Technical Report NIST 800-55, National Institute of Standards and Technology (2003)
3. Vaughn, Jr., R.B., Henning, R., Siraj, A.: Information assurance measures and metrics - state of practice and proposed taxonomy. In: Proceedings of the 36th Hawaii International Conference on Systems Sciences. (2003)
4. Bouvier, P., Longeon, R.: Le tableau de bord de la sécurité du système d'information. Sécurité Informatique (2003)
5. Nielsen, F.: Approaches of security metrics. Technical report, NIST-CSSPAB (2000)
6. Payne, S.C.: A guide to security metrics. Technical report, SANS Institute (2001)
7. ACSA, ed.: Proceedings of the Workshop on Information Security System Scoring and Ranking, Williamsburg, Virginia (2001)
8. Colado, C., Franco, A.: Métricas de seguridad: una visión actualizada. SIC. Seguridad en Informática y Comunicaciones **57** (2003) 64–66
9. Swanson, M.: Security self-assessment guide for information technology systems. Technical Report NIST 800-26, National Institute of Standards and Technology (2001)
10. Calero, C., Martín-Albo, J., Piattini, M., Vallecillo, M.B..A., Cechich, A.: A survey on software component metrics. Submitted to ACM Computing Surveys (2003)
11. Fenton, N., Pfleeger, S.: Software Metrics: A Rigorous Approach. 2nd edn. Chapman Hall, London (1997)
12. Whitmire, S.: Object Oriented Design Measurement. Wiley, New York (1997)
13. Zuse, H.: A Framework of Software Measurement. Walter de Gruyter, Berlin (1998)
14. Poels, G., Dedene, G.: Distance-based software measurement: Necessary and sufficient properties for software measures. Information and Software Technology **42** (2000) 35–46
15. Weyuker, E.J.: Evaluating software complexity measures. IEEE Transactions on Software Engineering **14** (1988) 1357–1365
16. Briand, L.C., Morasca, S., Basili, V.R.: Property-based software engineering measurement. IEEE Transactions on Software Engineering **22** (1996) 68–86
17. Briand, L.C., Morasca, S., Basili, V.R.: Property-based software engineering measurement: Refining the additivity properties. IEEE Transactions on Software Engineering **23** (1997) 196–197

18. Juristo, N., Moreno, A.: Basics of Software Engineering Experimentation. Kluwer Academic Publishers (2001)
19. Wohlin, C., Runeson, P., Ohlsson, M., Regnell, B., Wesslen, .A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers (2000)
20. Yin, R.: Case Study Research: Design and Methods. 2nd edn. Applied Social Research Methods Series, vol 5 Sage Publications Inc, Thousand Oaks, CA (1994)
21. Pfleeger, S., Kitchenham, B.: Principles of survey research. Software Engineering Notes **26** (2001) 16–18
22. Lavazza, L.: Providing automated support for the gqm measurement process. IEEE Software **17** (2000) 56–62
23. Departament of the Air Force: AFI33-205. Information Protection Metrics and Measurements Program. (1997)
24. Calero, C., Piattini, M., Genero, M.: Empirical validation of referential integrity metrics. Information and Software Technology **43** (2001) 949–957
25. ISO: ISO 7498-2. Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture. (1989)
26. ISO/IEC: ISO/IEC TR 13335-1. Guidelines for the Management of IT Security. Part I: Concepts and Models of IT Security. (1996)
27. ISO/IEC: ISO/IEC 15408. Evaluation Criteria for IT Security. (1999)
28. ISO/IEC: ISO/IEC 17799. Code of Practice for Information Security Management. (2000)
29. King, G.: Best security practices: An overview. In: Proceedings of the 23rd National Information Systems Security Conference, Baltimore, Maryland, NIST (2000)
30. Marcelo, J.M.: Identificación y Evaluación de Entidades en un Método AGR. In: Seguridad de las Tecnologías de la Información. AENOR (2003) 69–103
31. McKnight, W.L.: What is information assurance? CrossTalk. The Journal of Defense Software Engineering (2002) 4–6
32. Schuedel, G., Wood, B.: Adversary work factor as a metric for information assurance. In: Procedings of the New Security Paradigm Workshop, Ballycotton, Ireland (2000) 23–30
33. Carnegie Mellon University Pittsburgh, Pennsylvania: SSE-CMM Model Description Document. 3.0 edn. (2003)
34. Vaughn, Jr., R.B., Siraj, A., Dampier, D.A.: Information security system rating and ranking. CrossTalk. The Journal of Defense Software Engineering (2002) 30–32

APPENDIX

This appendix presents, in tabular form, the metrics that we have surveyed for our analysis, and the dimensions and features assigned to each of them.

Metric information is displayed in columns. Column one is a sequence counter (1 to 57). Column two show the metric name and description, together with the reference to the article in which the metric was originally defined. Columns three to six show the dimensions assigned to the metric. Finally, columns seven to twelve display the values assigned to the metric features.

The values assigned to the cells of the columns three at six they have the following meaning:

- Column SO (Security Objective): C (Confidentiality), I (Integrity), A (Availability), AU (Authentication) and G (General).
- Column CA (Control Area): M (Management), O (Operational) and T (Technical).

- Column TD (Temporal Dimension): P (Preventive), D (Detective), C (Corrective) and R (Recovery).
- Column IA (Intended Audience): T (Technical Experts), D (Decision makers) and E (External authorities).

The values assigned to the cells in the last two columns (Empirical validation and Automation) require some special explanations:

- Column ”EV” shows whether the metric has gone through any kind of empirical validation. Cells in this column may be either empty, or have the following values: ”1E” (validated by one experiment); or ”FW” (mentioned as future work by the metric authors).
- Column ”A” shows the kind of automated support for the metric. Cells of this column may be either empty, or have the value ”CT”, indicating that there is a tool that supports the metric. A description of such tool can be found in the paper cited for the metric.

N	Metric Description	SO	CA	TD	IA	O/S	D/I	R/S	TV	EV	A
1	Percentage of critical data files and operations with an established backup frequency [2]	A	O	P	T	O	I	S	N		
2	Percentage of systems that have a contingency plan [2]	A	O	P	E	O	I	S	N		
3	Percentage of systems for which contingency plans have been tested in the past year [2]	A	O	P	E	O	I	S	N		
4	Number of uses of the backups [4]	A	O	R	D	O	D	R	N		
5	Annual down time for a system [3]	A	T	D	D	O	D	R	N		
6	Percentage of systems that perform password policy verification [2]	AU	O	P	D	O	I	S	N		
7	Percentage of users with special access to systems who have undergone background evaluations [2]	AU	O	P	D	O	I	S	N		
8	Number of failed login attempts [3]	AU	O	D	T	O	D	R	N		
9	Percentage of systems without active vendor-supplied passwords [2]	AU	T	P	T	O	I	S	N		
10	Percentage of unique user [2]	AU	T	P	T	O	I	S	N		
11	Percentage of websites with a posted privacy policy [2]	C	O	P	D	O	I	S	N		
12	Percentage of used media sanitized before reuse or disposal [2]	C	O	P	T	O	I	R	N		
13	Number of clock cycles per byte encrypted [3]	C	T	P	T	O	D	R	N		
14	Percentage of laptops with encryption capability for sensitive files [2]	C	T	P	T	O	I	S	N		
15	Frequency of the audits [4]	G	M	P	D	O	D	R	N		
16	Number of rules for security's politics [4]	G	M	P	D	O	D	S	N		
17	Level of maturity in developers' process [33]	G	M	P	D	O	D	S	N		
18	Percentage allocated for security program [3]	G	M	P	D	O	D	S	N		
19	Percentage of systems that have had risk levels reviewed by management [2]	G	M	P	D	O	I	S	N		
20	Percentage of systems recertified if security controls are added/modified after the system was developed [2]	G	M	P	D	O	I	S	N		
21	Percentage of systems that are operating under an Interim Authority to Operate (IATO) [2]	G	M	P	D	O	I	S	N		
22	Percentage of systems with approved system security plans [2]	G	M	P	D	O	I	S	N		
23	Risk assessment [30]	G	M	P	D	S	D	R	N		CT
24	Percentage of systems that had formal risk assessments performed and documented [2]	G	M	P	E	O	I	S	N		
25	Percentage of total systems for which security controls have been tested and evaluated in the past year [2]	G	M	P	E	O	I	S	N		
26	Percentage of systems that have the costs of their security controls integrated into the life cycle [2]	G	M	P	E	O	I	S	N		
27	Percentage of total systems that have authorized for processing following certification and accreditation [2]	G	M	P	E	O	I	S	N		
28	Percentage of current security plans [2]	G	M	P	E	O	I	S	N		
29	Assessment of the execution of the recovery plans [4]	G	M	R	D	S	D	R	N		
30	Percentage of information systems libraries that log the deposits and withdrawals of tapes [2]	G	O	P	T	O	I	S	N		
31	Percentage of data transmission facilities in the organization that have restricted access to authorized users [2]	G	O	P	T	O	I	S	N		
32	Percentage of software changes documented and approved through change request forms [2]	G	O	P	T	O	I	S	N		
33	Percentage of in-house applications with documentation on file [2]	G	O	P	T	O	I	S	N		
34	Percentage of users with access to security software that are not security administrators [2]	G	O	P	T	O	I	S	N		
35	Number of hours employed in formation [4]	G	O	P	D	O	D	R	N		
36	Percentage of formed people [4]	G	O	P	D	O	I	R	N		
37	Percentage of systems compliant with the separation of duties requirement [2]	G	O	P	D	O	I	S	N		
38	Percentage of systems that impose restrictions on system maintenance personnel [2]	G	O	P	D	O	I	S	N		
39	Percentage of systems with documented risk assessment reports [2]	G	O	P	D	O	I	S	N		
40	Number of incidents reported to FedCIRC, NIPC, and local law enforcement [2]	G	O	P	E	O	D	R	N		
41	Percentage of employees with significant security responsibilities who have received specialized training [2]	G	O	P	E	O	I	R	N		
42	Percentage of agency components with incident handling and response capability [2]	G	O	P	E	O	I	S	N		
43	The average time elapsed between vulnerability discovery and implementation of corrective action [2]	G	O	C	T	O	D	R	N		
44	Percentage of security-related user issues resolved immediately following the initial call [2]	G	O	C	D	O	I	R	N		
45	Number of detected attacks [4]	G	O	D	D	O	D	R	N		
46	Number of invalid packets rejected for a firewall [3]	G	T	P	T	O	D	R	N		
47	Number of elements dedicated to network security [4]	G	T	P	T	O	D	S	N		
48	Number of components with audit trail [4]	G	T	P	T	O	D	S	N		
49	Evaluation Assurance Level according Common Criteria for a system [27]	G	T	P	T	O	D	S	N		
50	Percentage of systems with the latest approved patches installed [2]	G	T	P	T	O	I	R	N		
51	Percentage of systems with automatic virus definition updates and automatic virus scanning [2]	G	T	P	T	O	I	R	N		
52	Percentage of systems running restricted protocols [2]	G	T	P	T	O	I	S	N		
53	Percentage of systems on which audit trails provide a trace of user actions [2]	G	T	P	T	O	I	S	N		
54	Adversary work factor [32]	G	T	P	D	O	D	R	N	1E	
55	Number of intrusion attempts reported [23]	G	T	D	D	O	D	R	N		
56	Number of reported successful intrusions with limited access or total access [23]	G	T	D	D	O	D	R	N		
57	Number of systems with functions of integrity in files [4]	I	T	P	T	O	D	S	N		

Security in the Management of Networks with SNMPv3

L. J. García Villalba, J. H. Ortiz Monedero and R. Paucar Curasma

Department of Computer Systems and Programming
Complutense University of Madrid (UCM)
Juan del Rosal, 8 – 28040 Madrid (Spain)
javiergv@sip.ucm.es

Abstract. This paper describes an experimental study of the security in the management or monitoring of information from a host or teams of networks through the protocol SNMP in version 3, that is characterized in regards to security authentication and access control. There is also developed an Management Information Base (MIB) in ASNI language which will be read and written using the SNMPv3 protocol in which is observed the authentication based on the user. Finally some configurations are illustrated and results obtained from the study.

1 Introduction

The proliferation of the data networks in the last decades, LANs as well as WANs, and the relationship between them makes the aspects relative to their control and management taken into account more and more each time, converting themselves into something to which all those responsible for the networks have to pay great attention.

Given that the natural tendency of any network is to grow, there are added new applications and users make use of the network, the management systems used need to be sufficiently flexible to be able to support the new elements that have been added without the necessity of making drastic changes in the network.

SNMP (Simple Network Management Protocol), in its different versions, is a group of network management applications that use the services offered by TCP/IP and that have become a standard. At the root of interests shown by the IAB (Internet Activities Board) is the finding of a management protocol that was valid for the network of the internet, with its necessities due to its large dimensions.

The SNMP protocol defines an interchange of network management information where in the most basic form exists a system manager and an agent through databases of information. This simplicity allowed deficiencies to be seen such as: problems in the transfer of large quantities of information, little or no security, as well as the weak mechanisms of authentication and privacy.

The capacities of SNMP for the basic management of the network are good. In 1993 SNMPv2 was introduced, which was revised in 1996. SNMPv2 was oriented to correct the capacities of transmission of large quantities of information, nevertheless this version continued without offering any solution in respect to security and privacy. Specifically, neither SNMPv1, nor SNMPv2 can authenticate the source of the management message, much less provide encryption for that message. In a management network where authentication does not exist or is not possible there are possibilities that unauthorized users could easily execute management functions or even worse spy information when it is past from an agent to system manager. Due to this, many implementations in SNMPv1/SNMPv2 are limited in their capacity to only reading, which as a consequence, reduces the utilities of control and monitoring of the network.

To correct this type of deficiencies, which are of such great importance due to the evolution of the internet in the market, a work group was formed to generate a series of standards that were proposed in RFCs 2271-2275 and whose result is SNMPv3 [1]. In these documents the specifications are defined for security and access control of the networks managed with SNMP, and include the functionalities of versions SNMPv1 and SNMPv2 respectively.

2 Fundamentals of SNMP

The fundamental to understand SNMP is to take into account three essential concepts that have the function of interchanging information. In figure 1 the components of a network management system are illustrated, which are a manager and an agent.

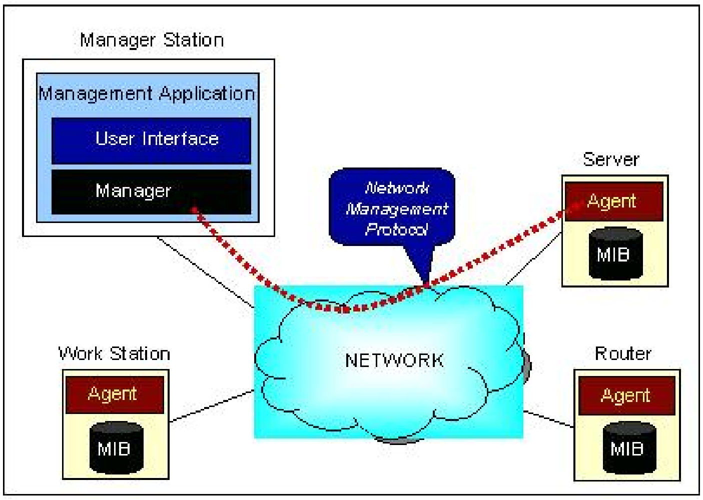


Fig. 1: Network Management System with SNMP.

In whatever configuration at least one management node has a software that supports SNMP. The management station generally provides an interface to the administrator of the network to control and observe the management processes in the network. This interface permits the user to execute commands (for example deactivate a link, read the IP address of one node and others) and provide general information of the system. The main point of the network management system is a group of applications that join the necessities in order to execute the functions. As a minimum a system will include basic applications to develop monitoring functions, configuration control and administration of the user accounts. More sophisticated systems may include more elaborate applications for these categories with more possibilities for the correction of errors.

On the other hand, the network devices when managed, including servers, workstations, personal computers, routers, etc. are equipped with a module that includes a software agent. The agent is responsible:

- To collect and maintain information about the local environment.
- To provide information to the user of the network, either in the form of an answer to a requirement or as an advisory message that abnormal something is happening.
- To respond to the commands executed by the user to change or alter the operation parameters or local configuration.

To execute these functions each agent maintains an MIB that contains all of the information (recent as well as historical) about its local configuration and the traffic that it manages. The management station will maintain a global MIB with the summarized information from all the agents.

It is important to high-light that all management applications generally share a common protocol in the entire network. This protocol provides the fundamental functions to request information and execute commands to the agents. This protocol, in our case SNMP, makes use of communication tools such as TCP/IP.

Specifically versions SNMPv1 and SNMPv2 consist of a group of documents that define a network management protocol, a general structure MIB and a specific member of MIB structured data for management purposes. In essence protocol provides four functions:

Get	Used by the manager to execute a requirement from an agent to an MIB.
Set	Used by the manager to change some value in an MIB from an agent.
Trap	Used by an agent to send an alert message to the manager.
Inform	Used by the manager to send an alert message to another manager.

3 SNMPv3

To correct the security deficiency that SNMPv1 and SNMPv2 have presented until now, a series of recommendations were written [2]. These recommendations are oriented to define an architecture and new capacities. SNMPv3 is a interoperable network management protocol, that provides access security to the devices by way of a combination of authentication and encryption of packages that travel by the network. The security capacities that SNMPv3 provides are:

Message Integrity	Assures that the package is not violated during transmission.
Authentication	Determines that the message comes from a valid source.
Encryption	Encrypts the contents of a package as a form of prevention.

3.1 Architecture

SNMPv3 [3] provides models as well as levels of security. A model of security is a strategy of authentication that is configured for the users and groups in which those reside. The levels of security refer to the level permitted to a user inside of a model of security. The combination of the two will determine which security mechanism will be used when an SNMP package is handled. SNMPv3 includes three services: authentication, privacy and access control. To provide these services in a sufficient form, SNMPv3 introduces a new concept called Main, the which is no more than an entity in the which the greater part of the services are proportioned or processed. A Main may act in an individual form or in a particular role, as an application or a group of applications or even as a combination of all of them. Essentially a Main operates from a management station and sends SNMP commands to the agents. The identity of the Main and that of the agent together determine the security capacity that will be invoked, including authentication, privacy and access control.

It is possible to define SNMPv3 in a modular form. Each SNMP entity includes a simple SNMP Engine. An SNMP Engine implements functions to send/receive, authenticate and encrypt/decrypt messages, in addition to controlling access to the handled objects. These functions are proportioned as services for one or more applications that are configured with the SNMP Engine to thus form the SNMP Entity as illustrated in the figure 2.

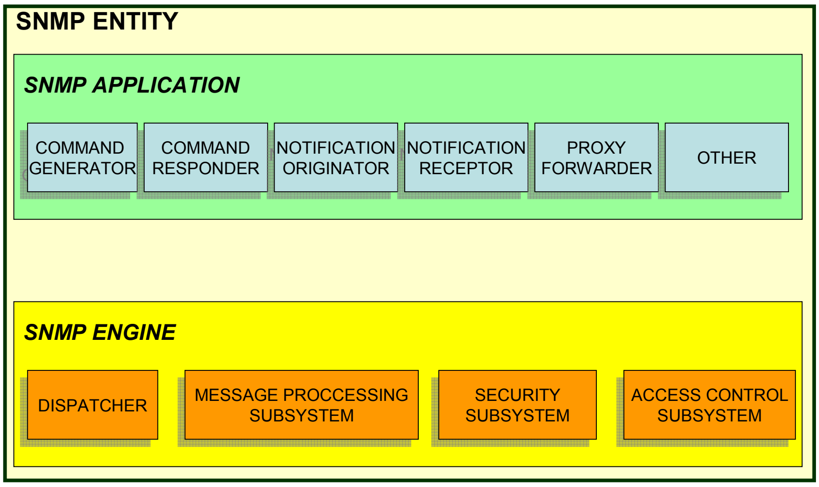


Fig. 2. Management Architecture of SNMPv3.

The modular architecture that is presented provides some advantages listed as follows:

- The role of the SNMP Entity is determined by modules that are implemented in that entity.
- The modular structure of the specifications allows the definition of different versions of each module, which makes it possible for them to take certain capacities and aspects of SNMP without the necessity of going to a new version

and taking the complete standard, in this way the co-existence of various versions are maintained.

3.2 Elements of a SNMP Entity

They are the following:

SNMP ENGINE:

Dispatcher. Permits the concurrence of multiple versions of SNMP messages in the SNMP Engine. It is responsible for:

- Accepting the PDUs (Protocol Data Units) from the applications so that later they are transmitted through the network, and sending the incoming PDUs to the applications.
- Passing the PDUs that leave to the Message Processing Subsystem so that they are prepared, and passing the incoming PDUs to the same subsystem so that they are extracted.
- Sending and receiving SNMP messages inside the network.

Message Processing Subsystem. Responsible for preparing messages to send and to extract the data of the received information.

Security Subsystem. Provides the services of authentication and privacy of the message. This subsystem potentially contains several security models.

Access Control Subsystem. Provides a set of services of authorization that an application can use for the control of access of the messages.

SNMP APPLICATION:

Command Generator. Starts the PDUs SNMP Get, GetNext, GetBulk or Set Request and processes the answer to a request that has been generated.

Command Responder. Receives the PDUs SNMP Get, GetNext, GetBulk or Set Request directed to the local system and later bring about the operation of the appropriate protocols using access control, and generates an answer message to be sent to the station that made the request.

Notification Originator. Monitors a system for a condition or a particular event and generates a Trap or an Inform message based on the condition or event. A notification originator should have a mechanism to determine where to send the message and which SNMP version and security parameters to use when the message is sent.

Notification Receptor. Waits for the notification messages and generates answers when a received message contains an Inform PDU.

Proxy Forwarder. Advances the SNMP messages. It is an optional application.

3.3 Message Processing

The model for message processing for SNMPv3 is generally defined in [3]. This model is responsible for accepting the PDUs from the dispatcher, encapsulates the messages and applies the USM (The User Security Model) [5] to insert the related parameters with the security in the heading of the message. The message processing model also takes charge of accepting incoming messages applying the USM to process the security parameters that are found in the heading of the message and sends the PDU to the dispatcher**.**

The structure of the message is illustrated in figure 4. The first five fields are generated by the incoming/outgoing message processing model. The following six fields show the security parameters used by the USM. Finally the PDU together with the ContextEngineID and ContextName constitute the PDU to be processed. The first five fields are the following:

msgVersion: Configured for SNMPv3.

msgID: An identifier used among the SNMP entities to coordinate request and answer messages. Its range is from 0 to $2^{31} – 1$.

MsgMaxSize. Refers to the maximum of a message in octets supported by the sender with a range of 484 to $2^{31}– 1$. This is the maximum size that a entity that sends can accept from another SNMP Engine.

MsgFlag. An array of octets that contains three flags in the three less significant bits.

- ReportableFlag: 1 is used for sent messages containing a request or an Inform and 0 is used for messages containing an answer Trap or Report PDU.
- PriorFlag and AuthFlag: Are configured by the sender to indicate the level of security applied to the message.

MsgSecurityModel. Is an identifier in the range of $2^{31} – 1$ that indicate the security model used by the sender, so that the receiver has the knowledge which security model he should use to process the message there exist reserved values: 1 for SNMPv1, 2 for SNMPv2, 3 for SNMPv3.

The six following fields related with the security parameters and generated by the USM include:

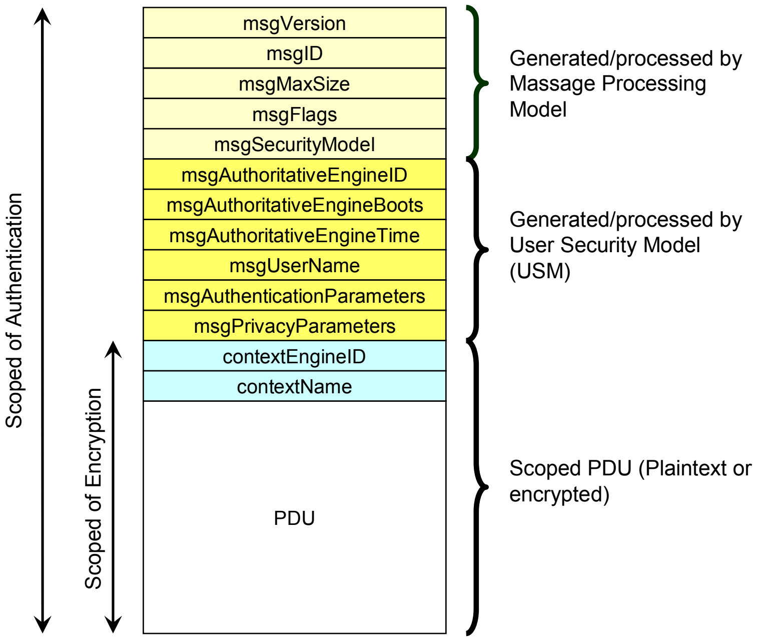
MsgAuthoritativeEngineID. Refers to the value of the source of a Trap, Response or Report and to the destination of a Get, GetNext, GetBulk, Set or Inform.

MsgAuthoritativeEngineTime. Is an whole value in the range of $2^{31} – 1$ that represents the number of seconds from when the snmpEngineBoots of the SNMP Engine was incremented.

MsgUserName. The principle user from which the message has been sent.

MsgAuthenticationParameters. Authentication Parameter. If the authentication is not used, this value is null. This parameters generated using an algorithm called HMAC.

MsgPrivacyParameters. Privacy Parameter. If the privacy is not used this value is null. This parameter is generated using an algorithm called DES.



Scoped of Authentication
Scoped of Encryption
msgVersion
msgID
msgMaxSize
msgFlags
msgSecurityModel
msgAuthoritativeEngineID
msgAuthoritativeEngineBoots
msgAuthoritativeEngineTime
msgUserName
msgAuthenticationParameters
msgPrivacyParameters
contextEngineID
contextName
PDU
Generated/processed by Massage Processing Model
Generated/processed by User Security Model (USM)
Scoped PDU (Plaintext or encrypted)

Fig. 3. Message Structure SNMP.

3.4 The Key for Authentication

The authentication mechanism in SNMPv3 assures that the message received was in reality transmitted by the principle entity source that appears in the identifying header of the message. In addition, this mechanism assures that the message was not altered during transmission and that it was not in some way delayed or captured and later resent by another source.

In the authentication process each principle and remote SNMP Engine that desires to communicate should share a secret authentication key. The entity that sends provides authentication including in the message a code. This code is a contained function of the message, of the SNMP Engine and of the principle about time of transmission and the secret key that should only be known by the sender and the receiver. The secret key should be configured initially by the administrator or user of the network, who will carry these keys in the data bases of the agents and the users. This can be done manually or using a secure form of data transmission.

When the receiving entity receives the message, it uses the same secret key to calculate the authentication code of the message. If the calculated code in the receiving side coincides with the value included with the sent message, then the receiver will know that the message originated from a authorized user and the message was not altered during transmission.

3.5 VACM (View-Based Access Control Model)

The Access Control Model makes possible the configuration of the agents to provide different access levels to the MIB and the different managers. An agent can restrict the access of its MIBs to one manager in particular in two ways: It can restrict the access to only certain parts of the MIB. The agent can limit the operation that a manager can use in certain portions of the MIB. The access control that is to be used by an agent for each manager should be pre-configured. It essentially consists of a table that details the access privileges of various authorized managers. The authentication differs in that it is done by the user, the access control is done by a group, where a group can be composed of a series of users. In figure 4 the logic of the functioning of this method of access control is illustrated

4 Implementation of the MIB Module for SNMPv3

The MIB development is done following the structure of the standard SMI [7] in one of the private nodes inside the intermediate node (internet) in figure 5 the structure of the nodes that represent the MIB created with its respective whole objects is illustrated [8]. The code written in the language ASN.1, contains the tree structure of the definition of the MIB. The final nodes indicate the whole objects that will be read or written. These will be accessed by the user of users configured in the agent that supports the protocol SNMPv3.

5 Conclusions

Now a days, the same as many other protocols used in internet, SNMP was found with the problems of security and privacy. For the which the SNMPv3 has the capacity of authentication, privacy, and access control of the information. Giving great confidence to the users of the market. As a change from the anterior versions, it is characterized by the level of security that is presented based in the user. It is for this that it has the necessity to create a user with its own respective key.

All of this was done in free distribution software such as Linux, using the SNMP package the which contains management tools, configuration files among others, being fundamental in the development of the MIB module.

Acknowledgements

Luis Javier García Villalba’s work is supported by the Spanish Ministry of Science and Technology (MCYT, Spain) under Project TIC2002-04516-C03-03. This author would like to express his appreciation to the Programa Complutense del Amo for providing him a grant to stay at IBM Research Division. During part of this work he

was with the Information Storage Group at the IBM Almaden Research Center, San Jose, California, USA (javiervi@almaden.ibm.com).

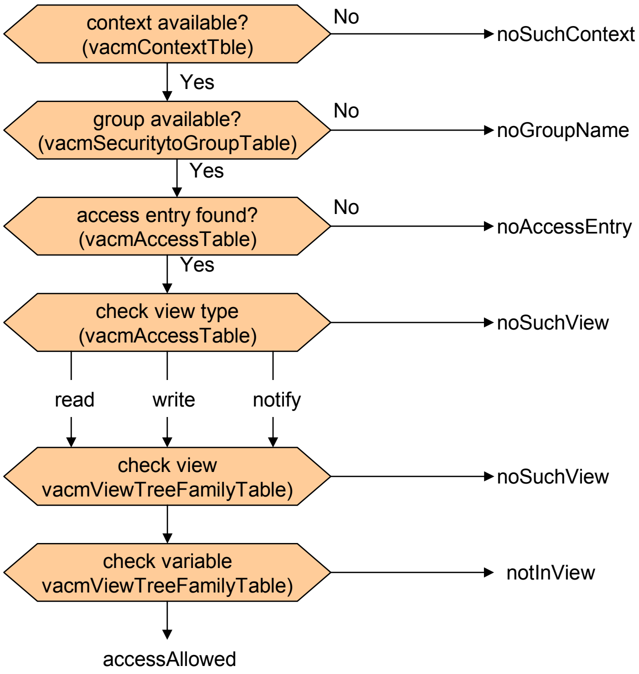


Fig. 4. Logic of the Functioning of Access Control.

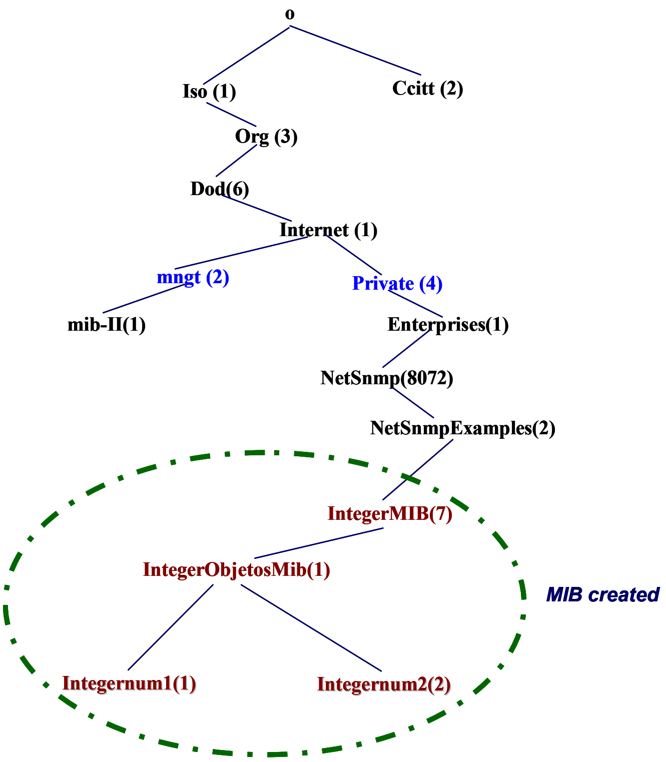


Fig. 5. Structure of the MIB Tree.

References

1. D. Harrington, R. Presuhn: "An Architecture for Describing SNMP Management Frameworks", IETF RFC 2271, January 1998.
2. William Stallings: "Security Comes to SNMP The New SNMPv3 Proposed Internet Standards", September 2001.
3. J. Case, D. Harrington: "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", IETF RFC 2272, January 1998.
4. D. Levi, P. Meyer: "SNMPv3 Applications", IETF RFC 2273, January 1998.
5. U. Blumenthal, B. Wijnen: "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", IETF RFC 2274, January 1998.
6. B. Wijnen, R. Presuhn: "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", IETF RFC 2275, January 1998.
7. M. Rose, K. McCloghrie: "Structure and Identification of Management Information for TCP/IP-based Internets", IETF RFC 1155, May 1990.
8. http://net-snmp.sourceforge.net/.

Author Index

INSTICC Press

Proceedings of the
2nd International Workshop on
Security In Information Systems, WOSIS 2004
ISBN: 972-8865-07-4
http://www.iceis.org